RAVE:
Installation and User Guide

Ian J. Grimstead

May 31, 2006

1 Installation Overview

Installation of clients is presented, which does not require a Web Service container. We follow with a Thin Client implemented in C++ (rather than Java, as used with the rest of RAVE); this is targeted at a PDA (namely a Sharp Zaurus). We then cover Data Service installation into Apache Axis within Jakarta Tomcat, followed by the similar Render Service installation.

2 Basic Client Installation: Java

The Thin Client does not require any rendering support (other than the basic Java J2SE default image handling libraries). It does, however, require Web Service support (such as XML, SOAP, UDDI handling). The current requirements are:

- axis.jar (from Apache Axis project, version 1.1, see http://ws.apache.org/axis/releases.html)
- commons-logging.jar (obtained from Apache Axis project, version 1.0.3, see Installing Apache Axis)
- commons-discovery.jar (obtained from Apache Axis project, version 0.2, see Installing Apache Axis)
- jaxrpc.jar (from Apache Axis project, version 1.1, see Installing Apache Axis)
- log4j-1.2.8.jar (obtained from Apache Axis project, version 1.2.8, see Installing Apache Axis)
- saaj.jar (obtained from Apache Axis project, version 1.1, see Apache Axis)
- uddi4j.jar (from UDDI4J project, version 2.0.2, see http://www.uddi4j.org)
- xercesImpl.jar (version provided with Apache Axis 1.1, see Installing Apache Axis)
- xmlParserAPIs.jar (version provided with Apache Axis 1.1, see Installing Apache Axis)

The obtaining of each jar file is now discussed in a separate section, followed by installing RAVE.

2.1 Installing Apache Axis

Download the binary distribution from http://ws.apache.org/axis/releases.html, namely `axis-1_1.tgz`. This archive contains a main subdirectory (`axis-1_1`), containing documentation (JavaDoc), jar files (which we need), sample applications and files to install into Jakarta Tomcat to support Web Services.

From the `lib` subdirectory, we need:

- axis.jar
- commons-logging.jar
- commons-discovery.jar
- jaxrpc.jar
- log4j-1.2.8.jar
- saaj.jar
- xercesImpl.jar
- xmlParserAPIs.jar

Copy these jar files into a common directory; for instance, we'll create a directory called `RAVE` and copy the files into it. We will copy all our jars into this directory, to simplify classpath creation.

2.2 Installing UDDI4J

Download the binary distribution from http://www-124.ibm.com/developerworks/oss/uddi4j/, namely `uddi4j-bin-2_0_2.zip`. Extract this archive (zipfile), this will create a main subdirectory (`uddi4j`), containing JavaDocs, sample code, sources and the UDDI4J jar file.

Copy from the `lib` subdirectory:

- uddi4j.jar

Copy this file into the previous common directory (we called ours `RAVE`).

2.3 Installing RAVE

Simply copy the RAVE jar file into the common directory (such as `RAVE-0.4-alpha.jar`).

3 Using the RAVE Service Management GUI to create a Thin Client

In this section, we present the RAVE Service Management GUI, explaining how to use it to generate a new collaboration session for use with a Thin Client. A Thin Client is so-called as it does not have local rendering facilities, so must make use of remote resources to carry out the rendering. In this case, we require a Data Service to host the data (ensuring all collaborating clients are synchronised), and a Render Service to render the data for the Thin Client (taking the data from the Data Service, rendering it to an off-screen frame buffer and presenting the output to the Thin Client).

3.1 Running RAVE Service Management GUI

The main client application for RAVE is the Service Management GUI; this discovers available RAVE services and instances, permitting the user to joint a pre-existing instance or to create a new instance.

To start the GUI, run `java`, inserting the jar files from this directory into the classpath. For example:

```
java -cp axis.jar:commons-discovery.jar:commons-logging.jar:jaxrpc.jar\
:log4j-1.2.8.jar:saaj.jar:uddi4j.jar:xercesImpl.jar:xmlParserAPIs.jar\
:RAVE-0.4-alpha.jar uk.ac.cardiff.cs.rave.net.ws.client.GUI.ServicePanelJ2SE
```

For speed, copy this line into a shell script (e.g. `run-rave-gui.sh`). This will launch the RAVE Web Services Client GUI, from which a Thin Client may in turn be launched.

```
[scmijg@adrenochrome RAVE]$ ./run-rave-gui.sh
Warning: Cannot convert string "-b&h-lucida-medium-r-normal-sans-*-140-*-*-p-*-iso8859-1" to type Fo
- RAVE version=1:1820
- RAVE build tag=0.4-alpha SNAPSHOT
- Using UDDI inquiry URL=[http://uddi.wesc.ac.uk:8334/juddi/inquiry]
- Using UDDI publish URL=[http://uddi.wesc.ac.uk:8334/juddi/publish]
- Finding RAVE business...
- RAVE Business ID: 7EDF6D20-1222-11D9-B11C-8E608C4352A8
- Finding RAVE tech model for RAVE Data Service...
- RAVE RAVE Data Service: Tech Model ID: uuid:7F2E0250-1222-11D9-B11C-9B79FB8F86F0
- Finding RAVE service for RAVE Data Service...
- Found access points for [RAVE Data Service]
- ...http://adrenochrome:8080/axis/services/RaveData
- ...http://ygrid01.grid.cf.ac.uk:8080/axis/services/RaveData
- ...http://ygrid02.grid.cf.ac.uk:8080/axis/services/RaveData
- ...http://tower.dr-grim.co.uk:8080/axis/services/RaveData
- ...http://ygrid03.grid.cf.ac.uk:8080/axis/services/RaveData
.
.
.
```

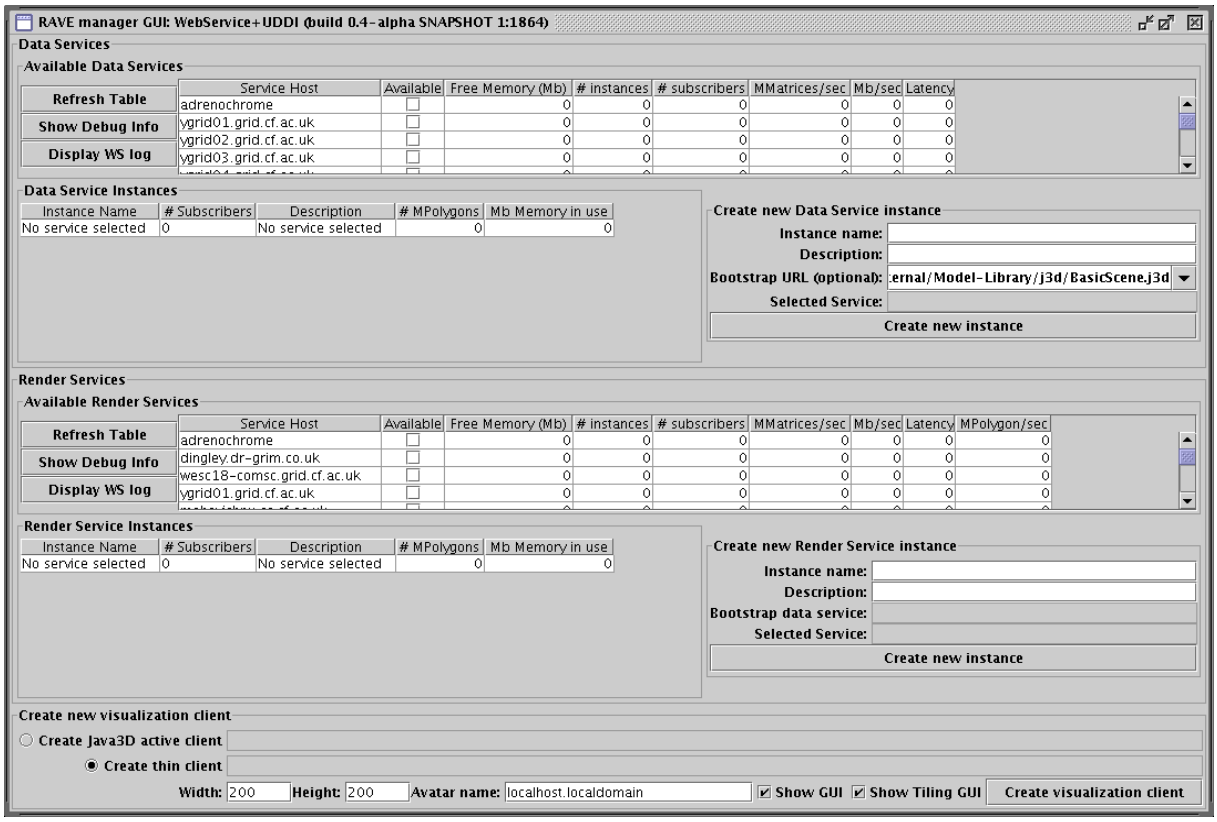

Figure 1: RAVE Service Manager GUI: Initial Screen

This shows the GUI starting up, and the UDDI server in WeSC being interrogated. A list of services is discovered, and the host machines are then queried for service status. This is used to populate the GUI, as presented in Figure 1. We describe the contents of the GUI, dividing it into three main sections: available data service, available render services and client creation.

3.2 Available Data Services

These are listed as the top third of the screen, inside the `Data Services` grouping box. The `Available Data Services` table shows each machine running a Data Service, with various statistics about the RAVE service and its host:

Service Host Name of the machine hosting the RAVE service

Available Tick-box, ticked if the server is available and responding to RAVE requests

Free Memory (Mb) The amount of memory available to RAVE (often set by the size of the Java Virtual Machine containing RAVE's Java implementation)

 # instances Number of service instances running on the host

 # subscribers Number of users subscribing to the services

MMatrices/sec Rough benchmark to return current CPU speed (represents millions of 4×4 matrices multiplied per second)

Mb/sec Available network bandwidth between the machine running the GUI and the remote RAVE service

Latency Network latency between the machine running the GUI and the remote RAVE service

The table can be sorted by column; simply click on the column title to sort in increasing order, click again for decreasing order, and click again to remove the column sort. This can be useful when wanting to find a Data Service with a particular attribute, such as network bandwidth or available memory. Three buttons are present, just to the left of the Data Services table:

Refresh Table Forces the GUI to repopulate the table of available Data Services, reconnecting to each machine

Show Debug Info Shows additional fields for debugging purposes

Display WS Log Shows the Web Services log output from the selected machine—debug support

Beneath `Available Data Services` is the `Data Service Instances` grouping box; this shows information about the currently selected Data Service. If a service host is selected in the `Available Data Services` table, RAVE instances maintained by the host are listed. The details of each instance are:

Instance Name Unique name used to identify this instance

 # Subscribers Number of remote users and services using this instance

Description Human-legible description of the contents of this instance

 # MPolygons How many million polygons are in the instance

Mb Memory in use An approximation of how much memory (in Mb) is used by this instance

If a service host is not responding, Data Service Instances will list `Service Not Available`; if a service host is responding but does not have any instances, `No Instances` will be shown instead. Finally, if instances are hosted and the service host is responding, then these will be listed as expected.

To the right of `Data Service Instances` is the grouping box entitled `Create new Data Service instance`. This lets the user create a new instance on the service host selected in the `Available Data Services` table. We need a Data Server to host a data service instance for us to visualize a dataset. We'll start by selecting a host that is responding and has spare capacity. We have selected `safleprofi.grid.cf.ac.uk`, as show in Figure 2.

Compared to Figure 1, `Create new Data Service instance` now has the `Selected Service` field filled in with the URL of `safleprofi.grid.cf.ac.uk`—this occurred when the service host was selected. Enter a name for a test instance in the `Instance Name` box; enter "demo". Similarly enter the `Description` field for the instance. To start with a populated instance, we need to enter a `Bootstrap URL`. This is a text field, but also a drop-down menu. To save time, select an entry from the drop-down menu; use `http://users.cs.cf.ac.uk/I.J.Grimstead/RAVE/internal/Model-Library/obj/galleon.obj`. This is a sample Wavefront format model of a galleon, provided with Sun's Java3D development kit.

Your GUI should now look like Figure 3; click on the `Create new instance` button to create the instance. A small window will pop-up to inform you that the creation has succeeded (as Figure 4). The GUI will now refresh to show you the new service instance details (Figure 5).

3.3 Available Render Services

These are listed in the middle third of the screen; each machine listed is running a RAVE Render Service, with the same statistics as described above with Data Services. Render Services have an additional field:

MPolygon/sec Rendering benchmark, shows millions of untextured polygons per second rendered off-screen by the Render Service

This enables the user to sort Render Services by the speed of render—useful if a high frame rate is required. The `Render Service Instances` grouping box shows similar detail as that described above for Data Services (except this time, the instance information is for the selected Render Service). Search for a Render Service host that has a good network bandwidth (`Mb/sec` field) to your local host; as an example, we've again selected `safleprofi.grid.cf.ac.uk`.

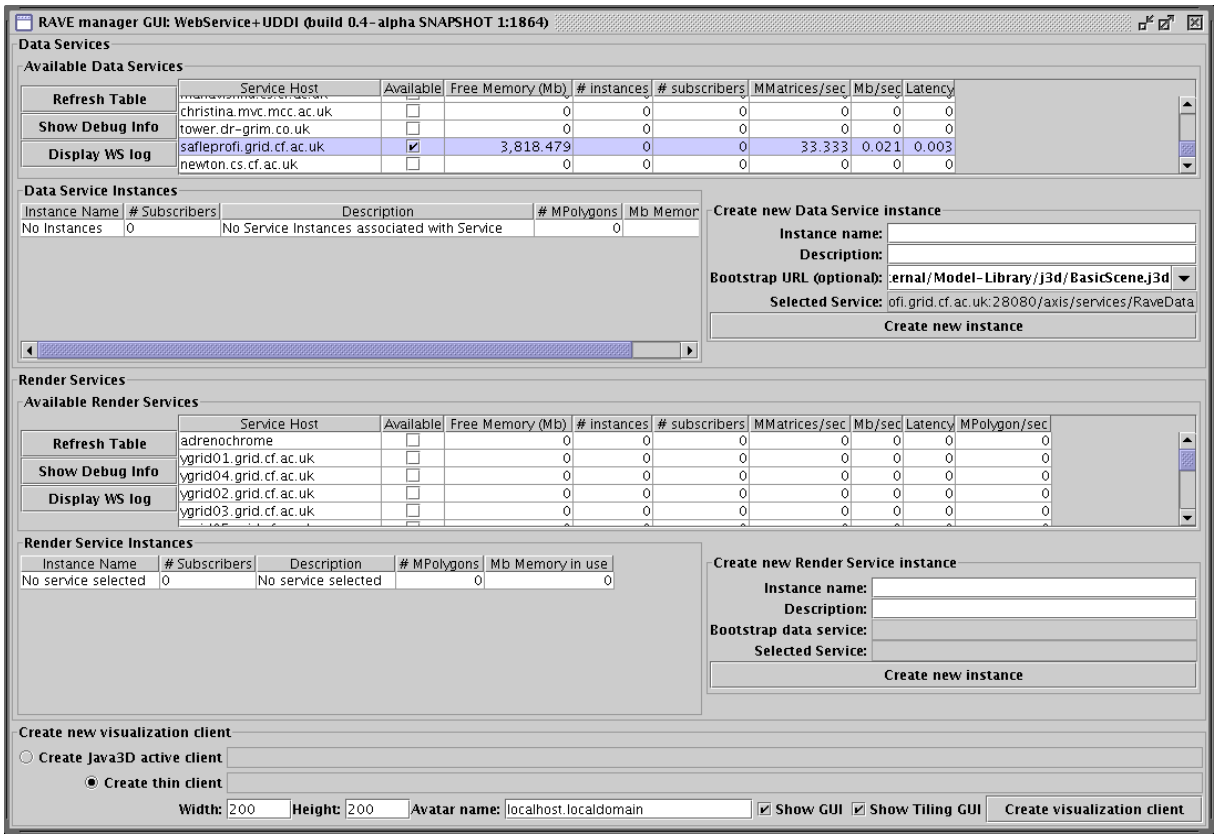
RAVE manager GUI: WebService+UDDI (build 0.4-alpha SNAPSHOT 1:1864)
Data Services
Available Data Services
Refresh Table
Show Debug Info
Display WS log
Service Host | Available | Free Memory (Mb) | # instances | # subscribers | MMatrices/sec | Mb/sec | Latency
christina.mvc.mcc.ac.uk | 0 | 0 | 0 | 0 | 0 | 0
tower.dr-grim.co.uk | 0 | 0 | 0 | 0 | 0 | 0
safleprofi.grid.cf.ac.uk | 3,818.479 | 0 | 0 | 33.333 | 0.021 | 0.003
newton.cs.cf.ac.uk | 0 | 0 | 0 | 0 | 0 | 0
Data Service Instances
Instance Name | # Subscribers | Description | # MPolygons | Mb Memor
No Instances | 0 | No Service Instances associated with Service | 0
Create new Data Service instance
Instance name:
Description:
Bootstrap URL (optional): ernal/Model-Library/j3d/BasicScene.j3d
Selected Service: ofi.grid.cf.ac.uk:28080/axis/services/RaveData
Create new instance
Render Services
Available Render Services
Refresh Table
Show Debug Info
Display WS log
Service Host | Available | Free Memory (Mb) | # instances | # subscribers | MMatrices/sec | Mb/sec | Latency | MPolygon/sec
adrenochrome | 0 | 0 | 0 | 0 | 0 | 0 | 0
ygrid01.grid.cf.ac.uk | 0 | 0 | 0 | 0 | 0 | 0 | 0
ygrid04.grid.cf.ac.uk | 0 | 0 | 0 | 0 | 0 | 0 | 0
ygrid02.grid.cf.ac.uk | 0 | 0 | 0 | 0 | 0 | 0 | 0
ygrid03.grid.cf.ac.uk | 0 | 0 | 0 | 0 | 0 | 0 | 0
Render Service Instances
Instance Name | # Subscribers | Description | # MPolygons | Mb Memory in use
No service selected | 0 | No service selected | 0 | 0
Create new Render Service instance
Instance name:
Description:
Bootstrap data service:
Selected Service:
Create new instance
Create new visualization client
Create Java3D active client
Create thin client
Width: 200 Height: 200 Avatar name: localhost.localdomain
Show GUI
Show Tiling GUI
Create visualization client

Figure 2: safleprofi.grid.cf.ac.uk Selected to Host a Data Service

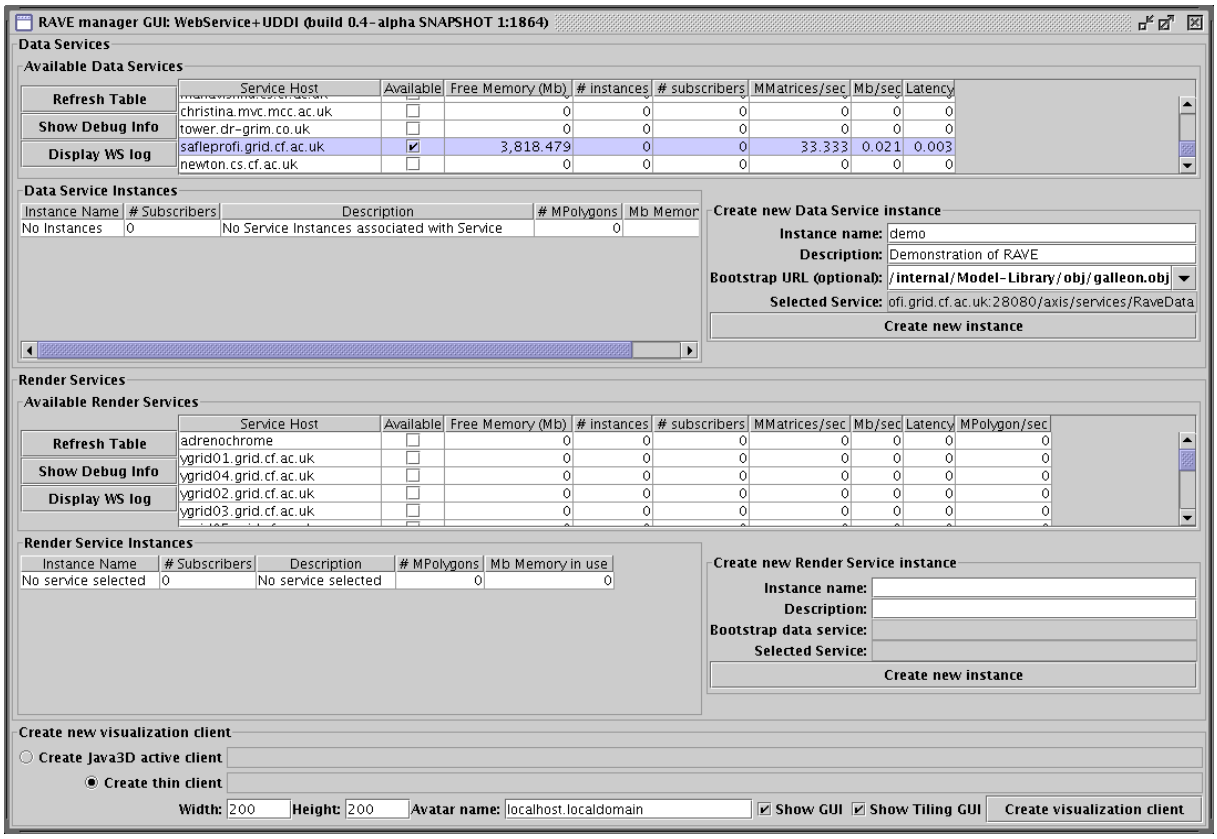
RAVE manager GUI: WebService+UDDI (build 0.4-alpha SNAPSHOT 1:1864)
Data Services
Available Data Services
Refresh Table
Show Debug Info
Display WS log
Service Host | Available | Free Memory (Mb) | # instances | # subscribers | MMatrices/sec | Mb/sec | Latency
christina.mvc.mcc.ac.uk | 0 | 0 | 0 | 0 | 0 | 0
tower.dr-grim.co.uk | 0 | 0 | 0 | 0 | 0 | 0
safleprofi.grid.cf.ac.uk | 3,818.479 | 0 | 0 | 33.333 | 0.021 | 0.003
newton.cs.cf.ac.uk | 0 | 0 | 0 | 0 | 0 | 0
Data Service Instances
Instance Name | # Subscribers | Description | # MPolygons | Mb Memor
No Instances | 0 | No Service Instances associated with Service | 0
Create new Data Service instance
Instance name: demo
Description: Demonstration of RAVE
Bootstrap URL (optional): /internal/Model-Library/obj/galleon.obj
Selected Service: ofi.grid.cf.ac.uk:28080/axis/services/RaveData
Create new instance
Render Services
Available Render Services
Refresh Table
Show Debug Info
Display WS log
Service Host | Available | Free Memory (Mb) | # instances | # subscribers | MMatrices/sec | Mb/sec | Latency | MPolygon/sec
adrenochrome | 0 | 0 | 0 | 0 | 0 | 0 | 0
ygrid01.grid.cf.ac.uk | 0 | 0 | 0 | 0 | 0 | 0 | 0
ygrid04.grid.cf.ac.uk | 0 | 0 | 0 | 0 | 0 | 0 | 0
ygrid02.grid.cf.ac.uk | 0 | 0 | 0 | 0 | 0 | 0 | 0
ygrid03.grid.cf.ac.uk | 0 | 0 | 0 | 0 | 0 | 0 | 0
Render Service Instances
Instance Name | # Subscribers | Description | # MPolygons | Mb Memory in use
No service selected | 0 | No service selected | 0 | 0
Create new Render Service instance
Instance name:
Description:
Bootstrap data service:
Selected Service:
Create new instance
Create new visualization client
Create Java3D active client
Create thin client
Width: 200 Height: 200 Avatar name: localhost.localdomain
Show GUI
Show Tiling GUI
Create visualization client

Figure 3: safleprofi.grid.cf.ac.uk Ready to Host a New Data Service Instance

Figure 4: safleprofi.grid.cf.ac.uk Popup on Creation of New Data Service Instance

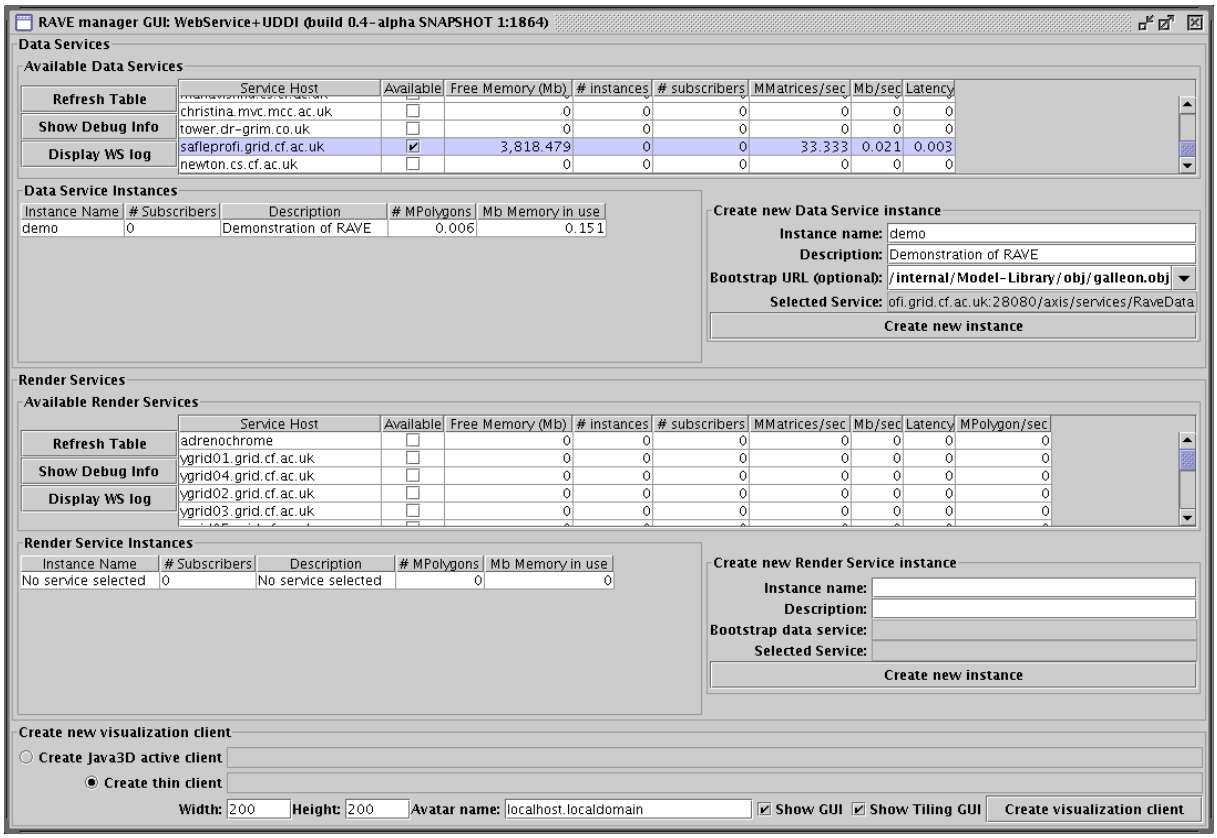
RAVE manager GUI: WebService+UDDI (build 0.4-alpha SNAPSHOT 1:1864)
Data Services
Available Data Services
Refresh Table
Show Debug Info
Display WS log
Service Host | Available | Free Memory (Mb) | # instances | # subscribers | MMatrices/sec | Mb/sec | Latency
christina.mvc.mcc.ac.uk | 0 | 0 | 0 | 0 | 0 | 0
tower.dr-grim.co.uk | 0 | 0 | 0 | 0 | 0 | 0
safleprofi.grid.cf.ac.uk | 3,818.479 | 0 | 0 | 33.333 | 0.021 | 0.003
newton.cs.cf.ac.uk | 0 | 0 | 0 | 0 | 0 | 0
Data Service Instances
Instance Name | # Subscribers | Description | # MPolygons | Mb Memory in use
demo | 0 | Demonstration of RAVE | 0.006 | 0.151
Create new Data Service instance
Instance name: demo
Description: Demonstration of RAVE
Bootstrap URL (optional): /internal/Model-Library/obj/galleon.obj
Selected Service: ofi.grid.cf.ac.uk:28080/axis/services/RaveData
Create new instance
Render Services
Available Render Services
Refresh Table
Show Debug Info
Display WS log
Service Host | Available | Free Memory (Mb) | # instances | # subscribers | MMatrices/sec | Mb/sec | Latency | MPolygon/sec
adrenochrome | 0 | 0 | 0 | 0 | 0 | 0 | 0
ygrid01.grid.cf.ac.uk | 0 | 0 | 0 | 0 | 0 | 0 | 0
ygrid04.grid.cf.ac.uk | 0 | 0 | 0 | 0 | 0 | 0 | 0
ygrid02.grid.cf.ac.uk | 0 | 0 | 0 | 0 | 0 | 0 | 0
ygrid03.grid.cf.ac.uk | 0 | 0 | 0 | 0 | 0 | 0 | 0
Render Service Instances
Instance Name | # Subscribers | Description | # MPolygons | Mb Memory in use
No service selected | 0 | No service selected | 0 | 0
Create new Render Service instance
Instance name:
Description:
Bootstrap data service:
Selected Service:
Create new instance
Create new visualization client
Create Java3D active client
Create thin client
Width: 200 Height: 200 Avatar name: localhost.localdomain
Show GUI
Show Tiling GUI
Create visualization client

Figure 5: GUI Showing New Data Service Instance Hosted by safleprofi.grid.cf.ac.uk

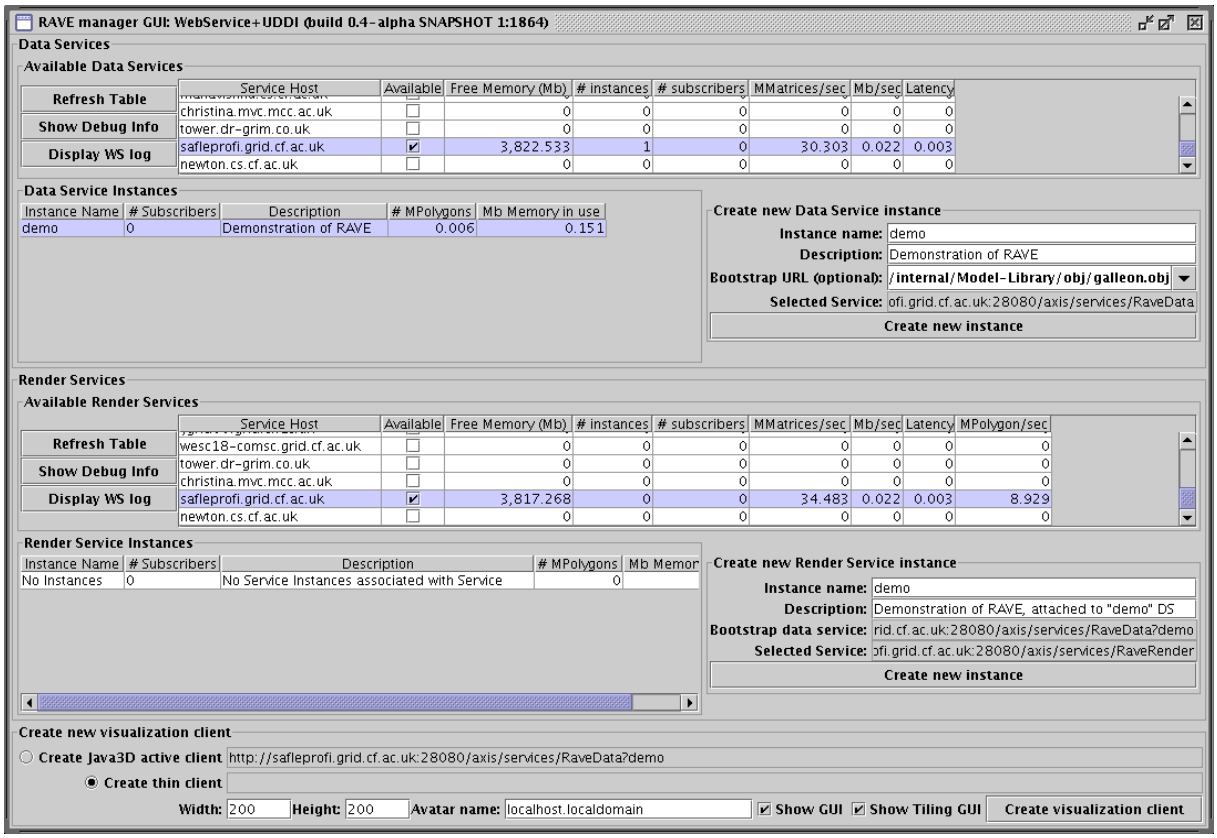
RAVE manager GUI: WebService+UDDI (build 0.4-alpha SNAPSHOT 1:1864)
Data Services
Available Data Services
Refresh Table
Show Debug Info
Display WS log
Service Host | Available | Free Memory (Mb) | # instances | # subscribers | MMatrices/sec | Mb/sec | Latency
christina.mvc.mcc.ac.uk | 0 | 0 | 0 | 0 | 0 | 0
tower.dr-grim.co.uk | 0 | 0 | 0 | 0 | 0 | 0
safleprofi.grid.cf.ac.uk | 3,822.533 | 1 | 0 | 30.303 | 0.022 | 0.003
newton.cs.cf.ac.uk | 0 | 0 | 0 | 0 | 0 | 0
Data Service Instances
Instance Name | # Subscribers | Description | # MPolygons | Mb Memory in use
demo | 0 | Demonstration of RAVE | 0.006 | 0.151
Create new Data Service instance
Instance name: demo
Description: Demonstration of RAVE
Bootstrap URL (optional): /internal/Model-Library/obj/galleon.obj
Selected Service: ofi.grid.cf.ac.uk:28080/axis/services/RaveData
Create new instance
Render Services
Available Render Services
Refresh Table
Show Debug Info
Display WS log
Service Host | Available | Free Memory (Mb) | # instances | # subscribers | MMatrices/sec | Mb/sec | Latency | MPolygon/sec
wesc18-comsc.grid.cf.ac.uk | 0 | 0 | 0 | 0 | 0 | 0 | 0
tower.dr-grim.co.uk | 0 | 0 | 0 | 0 | 0 | 0 | 0
christina.mvc.mcc.ac.uk | 0 | 0 | 0 | 0 | 0 | 0 | 0
safleprofi.grid.cf.ac.uk | 3,817.268 | 0 | 0 | 34.483 | 0.022 | 0.003 | 8.929
newton.cs.cf.ac.uk | 0 | 0 | 0 | 0 | 0 | 0 | 0
Render Service Instances
Instance Name | # Subscribers | Description | # MPolygons | Mb Memor
No Instances | 0 | No Service Instances associated with Service | 0
Create new Render Service instance
Instance name: demo
Description: Demonstration of RAVE, attached to "demo" DS
Bootstrap data service: rid.cf.ac.uk:28080/axis/services/RaveData?demo
Selected Service: ofi.grid.cf.ac.uk:28080/axis/services/RaveRender
Create new instance
Create new visualization client
Create Java3D active client http://safleprofi.grid.cf.ac.uk:28080/axis/services/RaveData?demo
Create thin client
Width: 200 Height: 200 Avatar name: localhost.localdomain Show GUI Show Tiling GUI Create visualization client

Figure 6: safleprofi.grid.cf.ac.uk Ready to Host a New Render Service Instance

Create Render Service Instance Succeeded
Creation of render service instance 'demo' in progress at
http://safleprofi.grid.cf.ac.uk:28080/axis/services/RaveRender
OK

Figure 7: safleprofi.grid.cf.ac.uk Popup on Creation of New Data Service Instance

To the right is the grouping box entitled `Create new Render Service Instance`; the `Instance name` and `Description` fields are again entered by hand, so for this demo we'll enter "demo" and "Demonstration Render Service" respectively.

RAVE Render Services bootstrap from a remote RAVE Data Service, so instead of entering a URL of a data source, we now need the URL of a RAVE Data Service instead. You cannot enter this by hand, you simply select the Data Service instance you wish to connect to in the `Data Service Instances` table. This enters the URL into the `Bootstrap URL` field. Finally, you must select which RAVE Render Service is to host this new service instance; select a Render Service host in the `Available Render Services` table. Your GUI should now match Figure 6.

Click the `Create new instance` button to create the Render Service instance; this will produce a pop-up window as shown in Figure 7. This is slightly different as when creating a Data Service (Figure 4), as Render Services bootstrap asynchronously with the RAVE Service Manager GUI; hence the message "In Progress". Once the bootstrap is complete, the GUI will refresh and will look like Figure 8. The new Render Service instance is listed, but also note that the Data Service has been updated—there is now one instance and one subscriber.

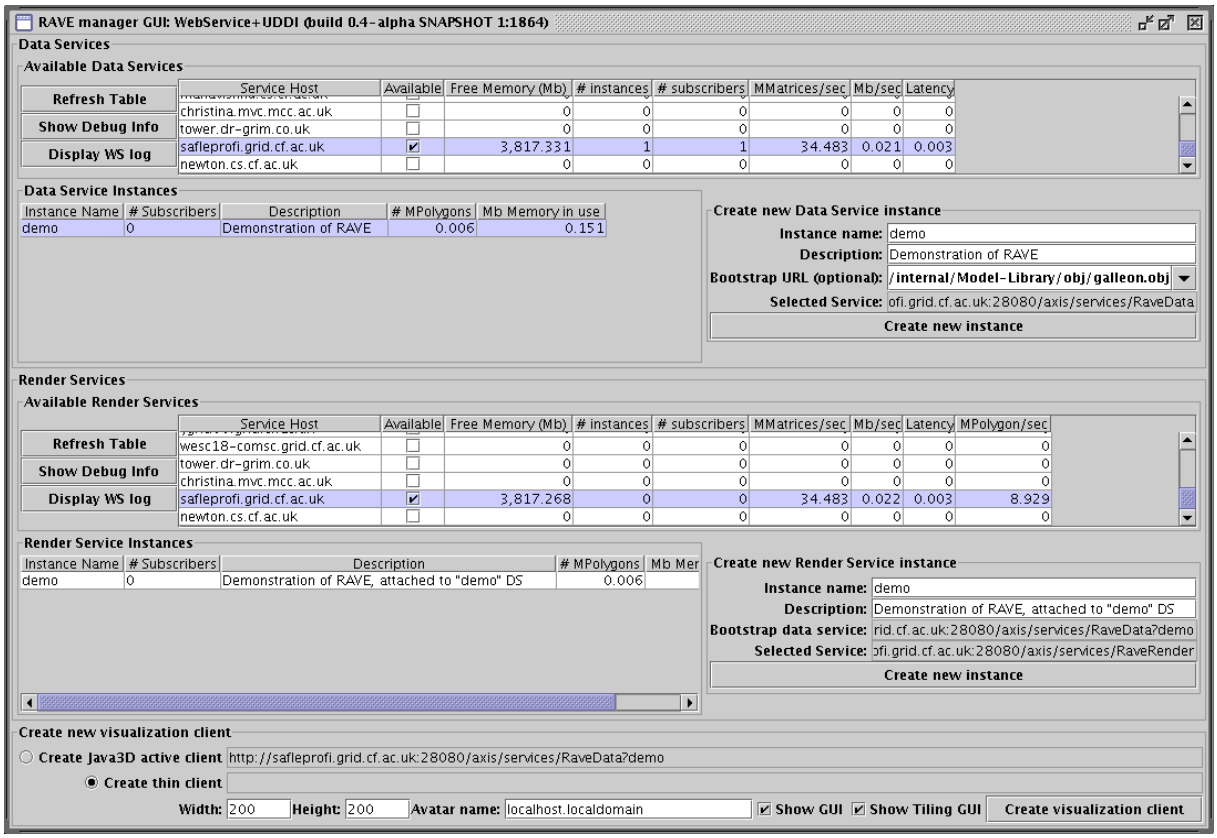
RAVE manager GUI: WebService+UDDI (build 0.4-alpha SNAPSHOT 1:1864)
Data Services
Available Data Services
Refresh Table
Show Debug Info
Display WS log
Service Host | Available | Free Memory (Mb) | # instances | # subscribers | MMatrices/sec | Mb/sec | Latency
christina.mvc.mcc.ac.uk | 0 | 0 | 0 | 0 | 0 | 0
tower.dr-grim.co.uk | 0 | 0 | 0 | 0 | 0 | 0
safleprofi.grid.cf.ac.uk | 3,817.331 | 1 | 1 | 34.483 | 0.021 | 0.003
newton.cs.cf.ac.uk | 0 | 0 | 0 | 0 | 0 | 0
Data Service Instances
Instance Name | # Subscribers | Description | # MPolygons | Mb Memory in use
demo | 0 | Demonstration of RAVE | 0.006 | 0.151
Create new Data Service instance
Instance name: demo
Description: Demonstration of RAVE
Bootstrap URL (optional): /internal/Model-Library/obj/galleon.obj
Selected Service: ofi.grid.cf.ac.uk:28080/axis/services/RaveData
Create new instance
Render Services
Available Render Services
Refresh Table
Show Debug Info
Display WS log
Service Host | Available | Free Memory (Mb) | # instances | # subscribers | MMatrices/sec | Mb/sec | Latency | MPolygon/sec
wesc18-comsc.grid.cf.ac.uk | 0 | 0 | 0 | 0 | 0 | 0 | 0
tower.dr-grim.co.uk | 0 | 0 | 0 | 0 | 0 | 0 | 0
christina.mvc.mcc.ac.uk | 0 | 0 | 0 | 0 | 0 | 0 | 0
safleprofi.grid.cf.ac.uk | 3,817.268 | 0 | 0 | 34.483 | 0.022 | 0.003 | 8.929
newton.cs.cf.ac.uk | 0 | 0 | 0 | 0 | 0 | 0 | 0
Render Service Instances
Instance Name | # Subscribers | Description | # MPolygons | Mb Mer
demo | 0 | Demonstration of RAVE, attached to "demo" DS | 0.006
Create new Render Service instance
Instance name: demo
Description: Demonstration of RAVE, attached to "demo" DS
Bootstrap data service: rid.cf.ac.uk:28080/axis/services/RaveData?demo
Selected Service: ofi.grid.cf.ac.uk:28080/axis/services/RaveRender
Create new instance
Create new visualization client
Create Java3D active client http://safleprofi.grid.cf.ac.uk:28080/axis/services/RaveData?demo
Create thin client
Width: 200 Height: 200 Avatar name: localhost.localdomain Show GUI Show Tiling GUI
Create visualization client

Figure 8: GUI Showing New Render Service Instance Hosted by safleprofi.grid.cf.ac.uk

3.4 Thin Client Creation

We have a Render Service instance ready to supply a local Thin Client, so we can create the client instance. In the bottom third of the screen, the grouping box `Create new visualization client` enables us to create both Thin and Active Clients. Firstly, we must select the visualization session instance we wish to view; select the "demo" Render Service instance we created earlier. This will be automatically entered into the `Create thin client` field; select the radio button next to it—this means we will create a Thin Client when we start the creation process. Now enter a width and height of your desired visualization window in pixels; 400×400 is a good size, and should stress the network. We then enter the name we'd like displayed to other users; enter your name, or perhaps the machine you're connecting from. We'll enter "Demo user #1" as an example. Make sure `Show GUI` is ticked (otherwise, you will be enable to interact with the Thin Client). Then make sure `Show Tiling GUI` is not ticked (this is used with Active Clients). The GUI will now look the same as Figure 9. Finally, click on the `Create visualization client` button to create the Thin Client. We will describe how to operate the Thin Client in the next section.

4 Using a RAVE Thin Client

In the previous section, we created a RAVE Thin Client; in this section, we will explain how to use it to navigate the dataset. Note that the GUI we will present is identical across all RAVE clients—be they PDA or workstation, Thin or Active, implemented in C++ or Java.

4.1 The Thin Client GUI: Basic Interactions

Having created the Thin Client in the previous section, you will now be presented with a window akin to Figure 10.

Click on the `Action` pull-down menu, and select `+Z focus` as presented in Figure 11. To active any action in RAVE, we use a single method: select action from menu, click and drag pointer on the

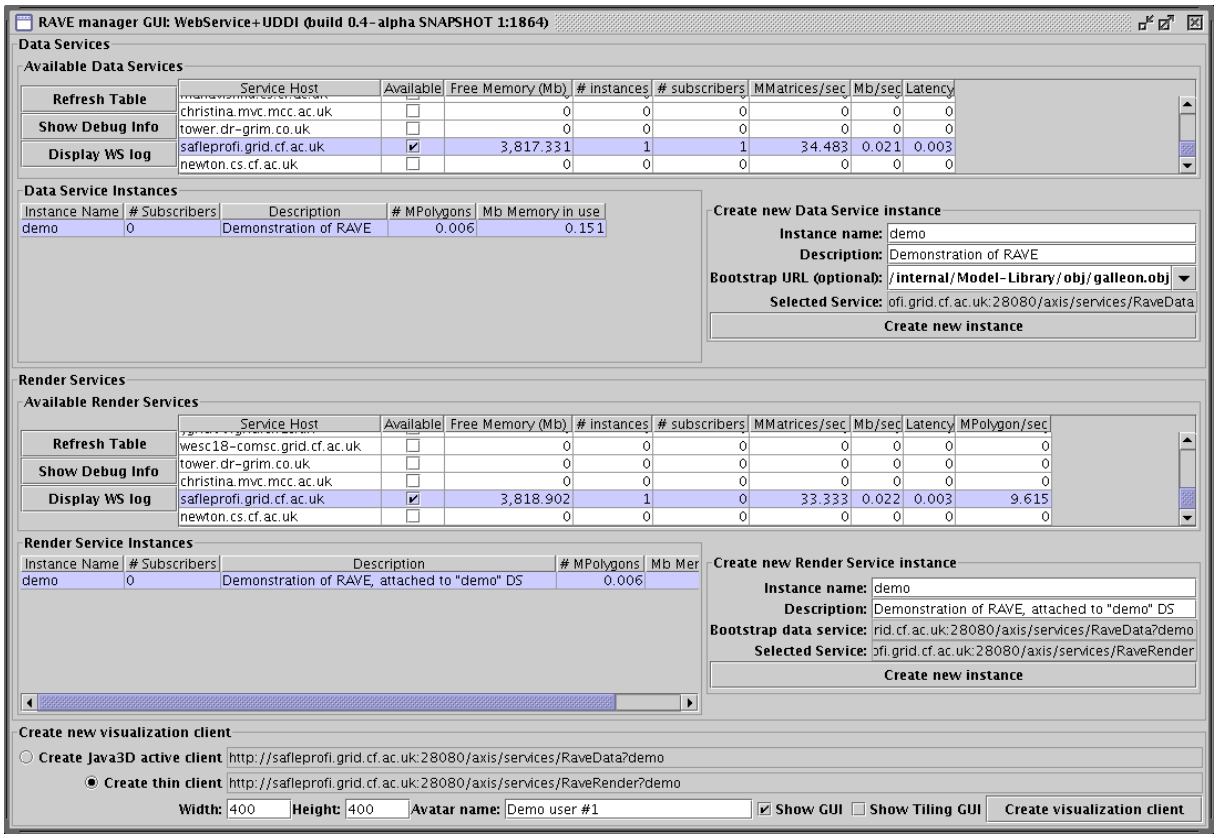
RAVE manager GUI: WebService+UDDI (build 0.4-alpha SNAPSHOT 1:1864)
Data Services
Available Data Services
Refresh Table
Show Debug Info
Display WS log
Service Host | Available | Free Memory (Mb) | # instances | # subscribers | MMatrices/sec | Mb/sec | Latency
christina.mvc.mcc.ac.uk | 0 | 0 | 0 | 0 | 0 | 0
tower.dr-grim.co.uk | 0 | 0 | 0 | 0 | 0 | 0
safleprofi.grid.cf.ac.uk | 3,817.331 | 1 | 1 | 34.483 | 0.021 | 0.003
newton.cs.cf.ac.uk | 0 | 0 | 0 | 0 | 0 | 0
Data Service Instances
Instance Name | # Subscribers | Description | # MPolygons | Mb Memory in use
demo | 0 | Demonstration of RAVE | 0.006 | 0.151
Create new Data Service instance
Instance name: demo
Description: Demonstration of RAVE
Bootstrap URL (optional): /internal/Model-Library/obj/galleon.obj
Selected Service: ofi.grid.cf.ac.uk:28080/axis/services/RaveData
Create new instance
Render Services
Available Render Services
Refresh Table
Show Debug Info
Display WS log
Service Host | Available | Free Memory (Mb) | # instances | # subscribers | MMatrices/sec | Mb/sec | Latency | MPolygon/sec
wesc18-comsc.grid.cf.ac.uk | 0 | 0 | 0 | 0 | 0 | 0 | 0
tower.dr-grim.co.uk | 0 | 0 | 0 | 0 | 0 | 0 | 0
christina.mvc.mcc.ac.uk | 0 | 0 | 0 | 0 | 0 | 0 | 0
safleprofi.grid.cf.ac.uk | 3,818.902 | 1 | 0 | 33.333 | 0.022 | 0.003 | 9.615
newton.cs.cf.ac.uk | 0 | 0 | 0 | 0 | 0 | 0 | 0
Render Service Instances
Instance Name | # Subscribers | Description | # MPolygons | Mb Mer
demo | 0 | Demonstration of RAVE, attached to "demo" DS | 0.006
Create new Render Service instance
Instance name: demo
Description: Demonstration of RAVE, attached to "demo" DS
Bootstrap data service: rid.cf.ac.uk:28080/axis/services/RaveData?demo
Selected Service: ofi.grid.cf.ac.uk:28080/axis/services/RaveRender
Create new instance
Create new visualization client
Create Java3D active client http://safleprofi.grid.cf.ac.uk:28080/axis/services/RaveData?demo
Create thin client http://safleprofi.grid.cf.ac.uk:28080/axis/services/RaveRender?demo
Width: 400 Height: 400 Avatar name: Demo user #1 Show GUI Show Tiling GUI Create visualization client

Figure 9: Ready to Create a RAVE Thin Client

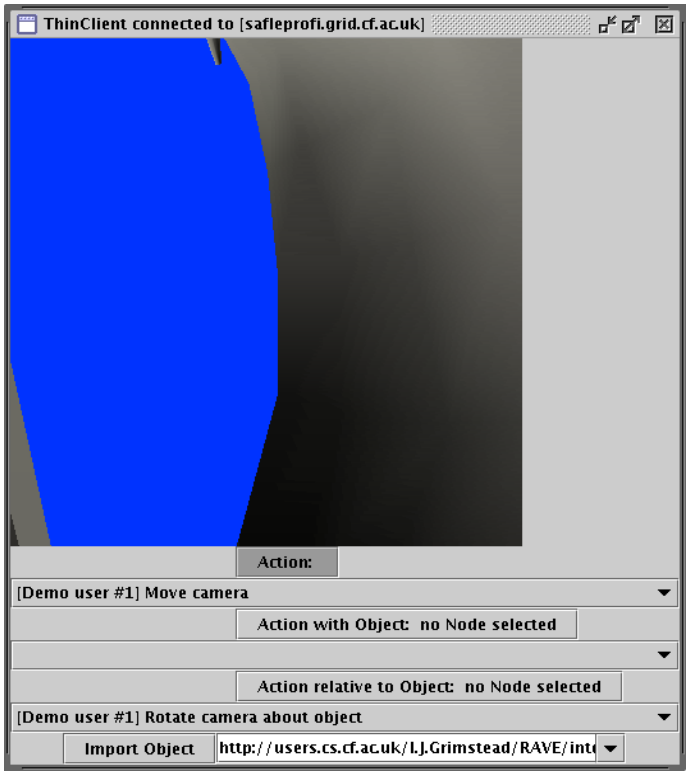
ThinClient connected to [safleprofi.grid.cf.ac.uk]
Action:
[Demo user #1] Move camera
Action with Object: no Node selected
Action relative to Object: no Node selected
[Demo user #1] Rotate camera about object
Import Object
http://users.cs.cf.ac.uk/I.J.Grimstead/RAVE/int

Figure 10: RAVE Thin Client—Initial View of the `galleon` Dataset

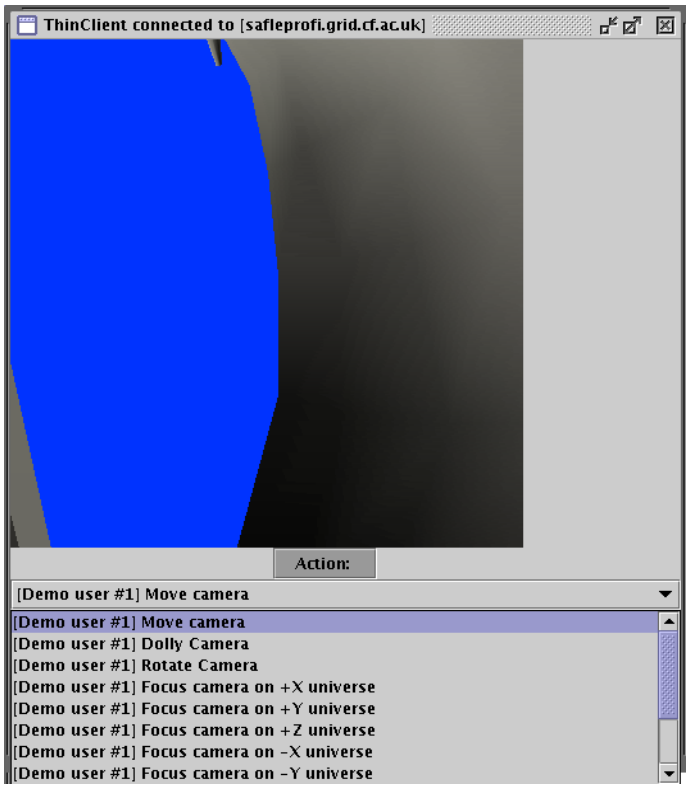
ThinClient connected to [safleprofi.grid.cf.ac.uk]
Action:
[Demo user #1] Move camera
[Demo user #1] Move camera
[Demo user #1] Dolly Camera
[Demo user #1] Rotate Camera
[Demo user #1] Focus camera on +X universe
[Demo user #1] Focus camera on +Y universe
[Demo user #1] Focus camera on +Z universe
[Demo user #1] Focus camera on -X universe
[Demo user #1] Focus camera on -Y universe

Figure 11: Select an Action to be Carried Out

image to active the action. The drag may be used to control the action, or simply to activate it. In this, case dragging simply activates the camera focus. The galleon is now visible, as shown in Figure 12.

An interactive action we can carry out is manually moving the camera; select `Dolly Camera` from the `Action` menu. Drag the pointer down the screen, and hold the mouse button down. The drag distance is interpreted as how fast to move the camera—it will continually dolly until the mouse button is released. This simulates leaning a joystick away from its position of rest, so the further you drag, the more dramatic the action. If you drag down, the camera will move forwards. If you drag up, the camera will move backwards.

To speed interaction, RAVE uses lossy, incremental frame encoding. This results in a low quality image during interaction; an example is presented in Figure 13. Note the sharp contrasts in colour on the sails, and the horizontal lines on the hull of the boat. These are artifacts from the lossy compression. However, once the dolly is complete (and the user stops moving the camera), the image will incrementally improve until the image is lossless, as in Figure 14.

You can now try the other supported actions; these are listed in the `Action:` drop-down menu, and include:

Move camera Mouse drags are interpreted as vertical or horizontal movement of the camera, in the image plane

Dolly camera Up and down mouse drags produce forwards and backwards movement (in the direction of the camera)

Rotate camera Rotate the camera about its X axis (with vertical mouse movement) and Y axis (with horizontal mouse movement)

Focus camera on $< axis >$ **universe** The camera will focus on the entire scene, from one of the 6 axis extremes (positive or negative X, Y or Z axis)

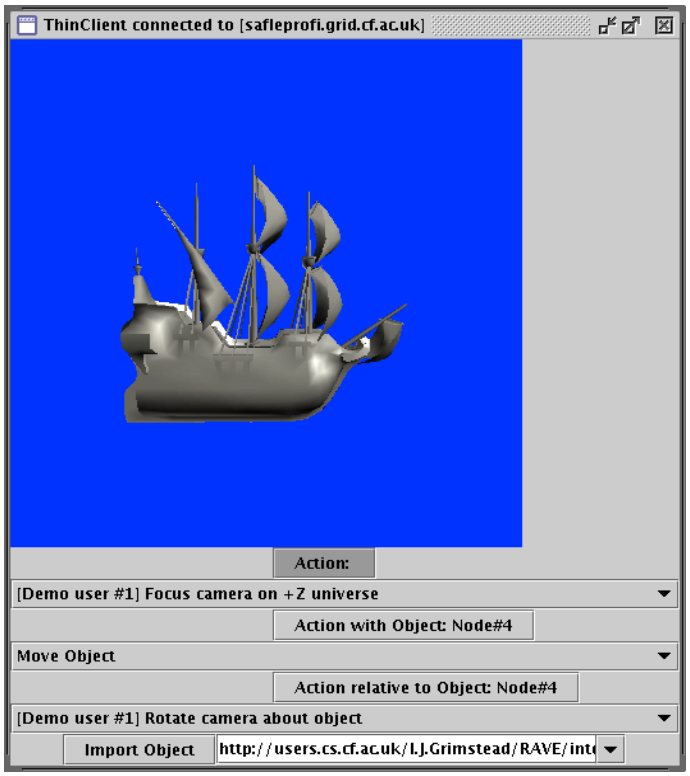

Figure 12: Focus on +Z Axis Completed

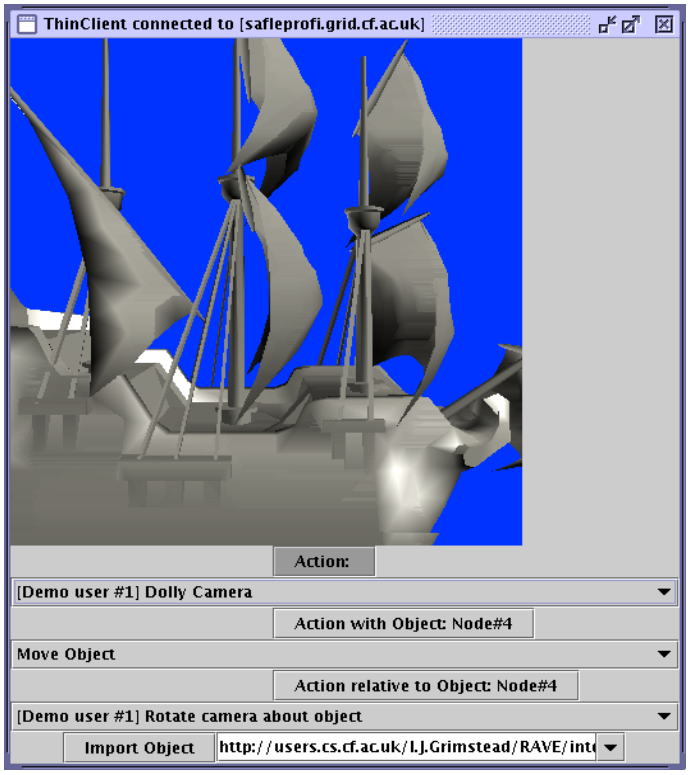

Figure 13: Dolly Camera Action in Progress—Showing Lossy Image

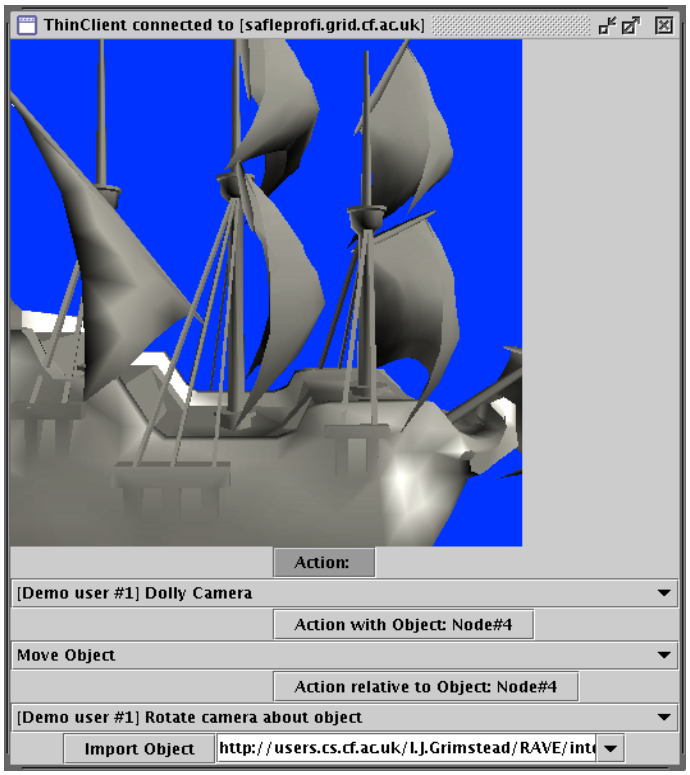

Figure 14: Dolly Camera Action Completed—Showing Lossless Image

4.2 Thin Client GUI: Actions Relative to Objects

The basic actions described above are carried out in isolation—namely, they only affect the user's camera (and their related avatar). Actions relative to objects require an object to be initially selected; to do this, click over the object in the view. If an object was selected, the GUI will be updated and enable the other two groups of actions:

Action with Object: $< id >$ Rotate or Move an object. Drag the pointer as to move the camera, only this will move an object instead

Action relative to Object: $< id >$ These vary according to the object selected, but include rotating the camera relative to an object (namely: orbiting an object)

By default, the `Action:` button is depressed; this means that any drag of the mouse will perform the selected action in the related drop-down menu. To perform an action with an object or relative to an object, select the desired option in the related drop-down menu, and then depress the related button (`Action with Object:` $< id >$ or `Action relative to object:` $< id >$). Dragging the mouse will now perform the selected operation.

Finally, there is the `Import Object:` $< url >$ button; this enables an additional scene to be imported into the scene. If you enter a URL into the text box (or select a URL from the drop-down menu), and click the button, the given scene will be imported into the RAVE instance at the position of the user's camera. The new dataset will be visible to all other users. Note that the scale of the object is not altered, so you may introduce an object so small you cannot see it, or an object so large that it shrouds the entire scene; be careful with this!

Note: some actions listed in the GUI may be work in progress, and hence may be unstable. Move and rotate camera or object are stable, as are focus camera on scene.
Add light, track camera, rotate around object and import object are unstable as of build *0.4-alpha*.

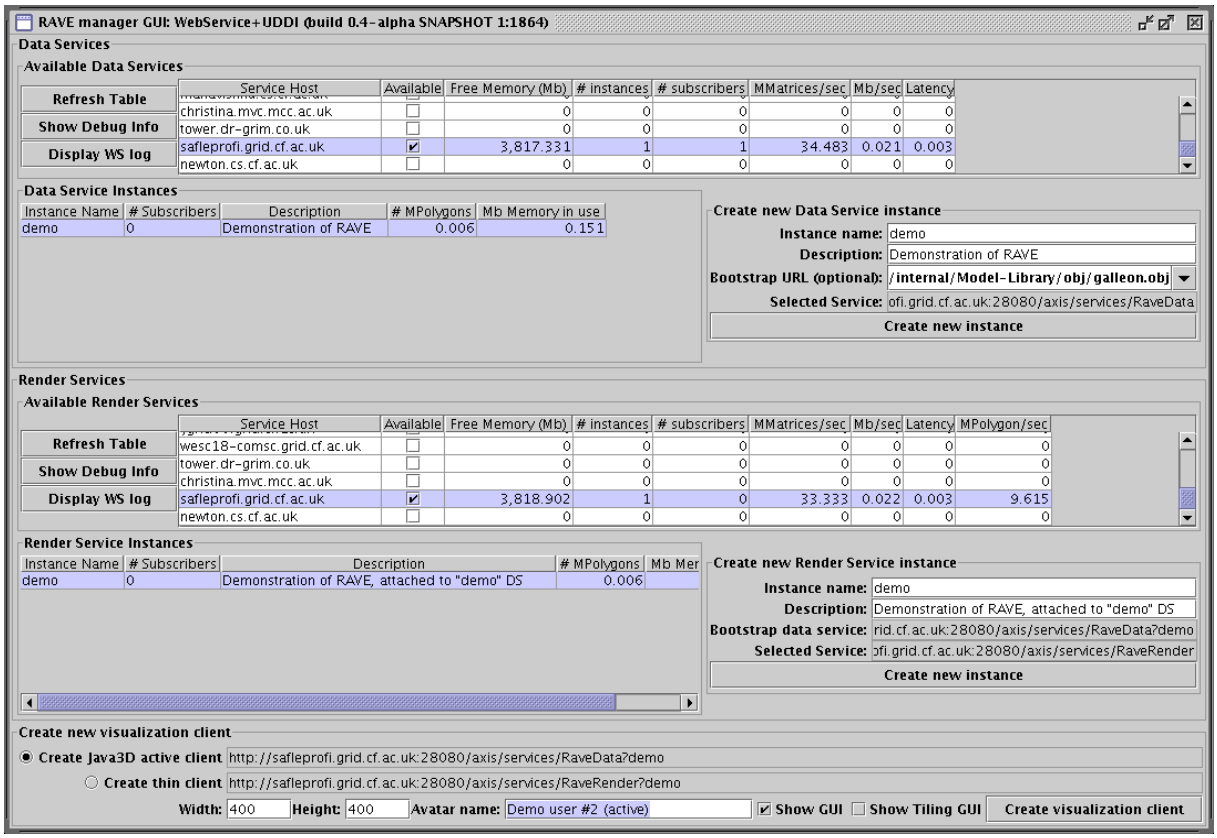
RAVE manager GUI: WebService+UDDI (build 0.4-alpha SNAPSHOT 1:1864)
Data Services
Available Data Services
Refresh Table
Show Debug Info
Display WS log
Service Host | Available | Free Memory (Mb) | # instances | # subscribers | MMatrices/sec | Mb/sec | Latency
christina.mvc.mcc.ac.uk | 0 | 0 | 0 | 0 | 0 | 0
tower.dr-grim.co.uk | 0 | 0 | 0 | 0 | 0 | 0
safleprofi.grid.cf.ac.uk | 3,817.331 | 1 | 1 | 34.483 | 0.021 | 0.003
newton.cs.cf.ac.uk | 0 | 0 | 0 | 0 | 0 | 0
Data Service Instances
Instance Name | # Subscribers | Description | # MPolygons | Mb Memory in use
demo | 0 | Demonstration of RAVE | 0.006 | 0.151
Create new Data Service instance
Instance name: demo
Description: Demonstration of RAVE
Bootstrap URL (optional): /internal/Model-Library/obj/galleon.obj
Selected Service: ofi.grid.cf.ac.uk:28080/axis/services/RaveData
Create new instance
Render Services
Available Render Services
Refresh Table
Show Debug Info
Display WS log
Service Host | Available | Free Memory (Mb) | # instances | # subscribers | MMatrices/sec | Mb/sec | Latency | MPolygon/sec
wesc18-comsc.grid.cf.ac.uk | 0 | 0 | 0 | 0 | 0 | 0 | 0
tower.dr-grim.co.uk | 0 | 0 | 0 | 0 | 0 | 0 | 0
christina.mvc.mcc.ac.uk | 0 | 0 | 0 | 0 | 0 | 0 | 0
safleprofi.grid.cf.ac.uk | 3,818.902 | 1 | 0 | 33.333 | 0.022 | 0.003 | 9.615
newton.cs.cf.ac.uk | 0 | 0 | 0 | 0 | 0 | 0 | 0
Render Service Instances
Instance Name | # Subscribers | Description | # MPolygons | Mb Mer
demo | 0 | Demonstration of RAVE, attached to "demo" DS | 0.006
Create new Render Service instance
Instance name: demo
Description: Demonstration of RAVE, attached to "demo" DS
Bootstrap data service: rid.cf.ac.uk:28080/axis/services/RaveData?demo
Selected Service: ofi.grid.cf.ac.uk:28080/axis/services/RaveRender
Create new instance
Create new visualization client
Create Java3D active client http://safleprofi.grid.cf.ac.uk:28080/axis/services/RaveData?demo
Create thin client http://safleprofi.grid.cf.ac.uk:28080/axis/services/RaveRender?demo
Width: 400 Height: 400 Avatar name: Demo user #2 (active)
Show GUI
Show Tiling GUI
Create visualization client

Figure 15: Ready to Create a RAVE Active Client

5 Using a RAVE Active Client

The Active Client requires local rendering support, so hence uses Java3D for rendering and various related libraries to support import/export of Java3D data.

5.1 Using the RAVE Service Management GUI to create an Active Client

The Active Client uses Java3D to render the scene, so you must first ensure that you have Java3D installed (version 1.3.1 or later). You may download this from https://java3d.dev.java.net/.

The Active Client will use more memory than the Thin Client, as the Thin Client's main storage requirement is its frame buffer, while the Active Client has to store the entire scene graph. To allocate more memory to the Java Virtual Machine (JVM), use the `-Xmx<memory in Mb>` option, viz:

```
java -Xmx300M -cp axis.jar:commons-discovery.jar:commons-logging.jar:jaxrpc.jar\
:log4j-1.2.8.jar:saaj.jar:uddi4j.jar:xercesImpl.jar:xmlParserAPIs.jar\
:RAVE-0.4-alpha.jar uk.ac.cardiff.cs.rave.net.ws.client.GUI.ServicePanelJ2SE
```

Restart the GUI using this script, which will allocate 300Mb for the JVM. Select `safleprofi.grid.cf.ac.uk` for the Data Service, and select the `galleon` dataset. Enter the size of the on-screen window in the `Create visualization client` grouping box, the name of the avatar (we'll use "Demo user #2 (active)"), make sure the tick box for `Show GUI` is ticked and `Show Tiling GUI` is not ticked (we will cover tiled rendering in a later section). Your GUI will now look similar to Figure 15.

Finally, click on the `Create visualization client` button to create the Thin Client. You will be presented with a GUI as in Figure 16.

Note that the number of subscribers to the Data Service will increment by one—there should now be two subscribers listed, one for the Active Client and one for the Render Service.

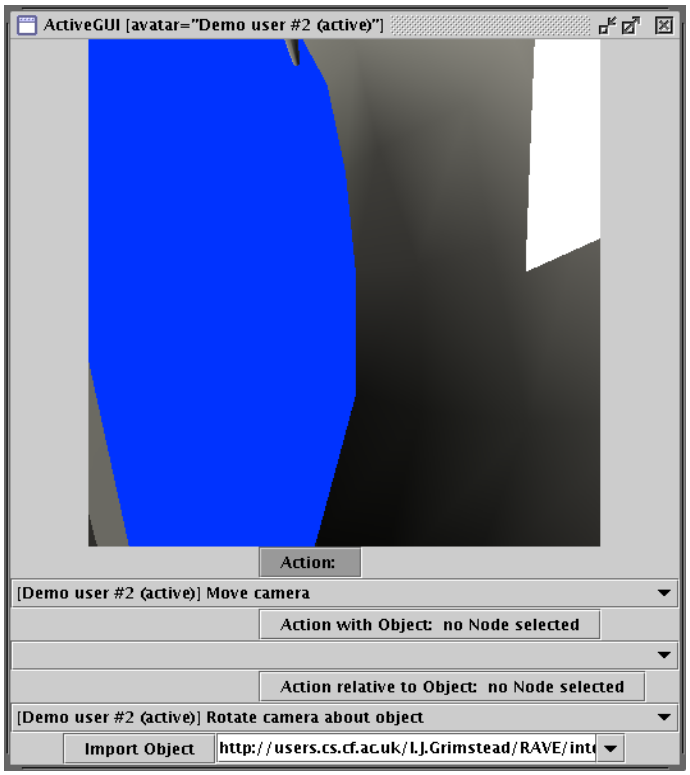
ActiveGUI [avatar="Demo user #2 (active)"]
Action:
[Demo user #2 (active)] Move camera
Action with Object: no Node selected
Action relative to Object: no Node selected
[Demo user #2 (active)] Rotate camera about object
Import Object
http://users.cs.cf.ac.uk/I.J.Grimstead/RAVE/int

Figure 16: RAVE Active Client—Initial View of the `galleon` Dataset

5.2 Interaction Between an Active Client and a Thin Client

If you now focus the camera on the Active Client on -ve Z (using the `Action` button) and then dolly the camera to get closer to the ship, you will be able to see the Thin Client's avatar as in Figure 17. Zooming still further (Figure 18) reveals the avatar representing the Thin Client.

Also note that the Active Client's avatar is now visible on the Thin Client. Examine Figure 19; a small white strip is visible behind the central mast, this is the name of the Active Client's avatar as seen from the Thin Client.

Zooming through the `galleon` with the Active Client eventually reveals the Thin Client's avatar as Figure 20, and the Active Client's avatar as seen from the Thin Client (Figure 21).

The Thin Client will see the Active Client's avatar moving on-screen, with latency dependant on the network between the local client and the remote RAVE service hosts. If the Thin Client rotates its camera, the Active Client will see the pointer underneath the Thin Client's avatar name rotate, see Figure 22.

6 Alternative RAVE Thin Client: C++ PDA Implementation

The Thin Client has also been implemented in C++, and is available as a `.ipk` file for installation on a Zaurus C750 PDA (see Figure 23). It is also available as a stand-alone executable for a workstation, but not documented here (as the RAVE Management GUI already contains a Thin Client for a workstation). The `.ipk` should also work on other models of Zaurus, such as C760, C860, C3000, 5500, etc.

6.1 Installing the Thin Client

Copy `RAVE-thinClient_0.4-alpha_arm.ipk` onto your Zaurus, either over the network or via the synchronisation tool. Copy the `.ipk` into `/home/zaurus/Documents/Install_Files`, then select the `Settings` tab on the Zaurus. Click on the `Add/Remove Software` button, then the `Install packages` button; this will list `RAVE-thinClient` with version `0.4-alpha`. Install this file (either to `Internal Flash` or `SD Card`, either will work). A small window will appear with the message `Now installing`; wait until this

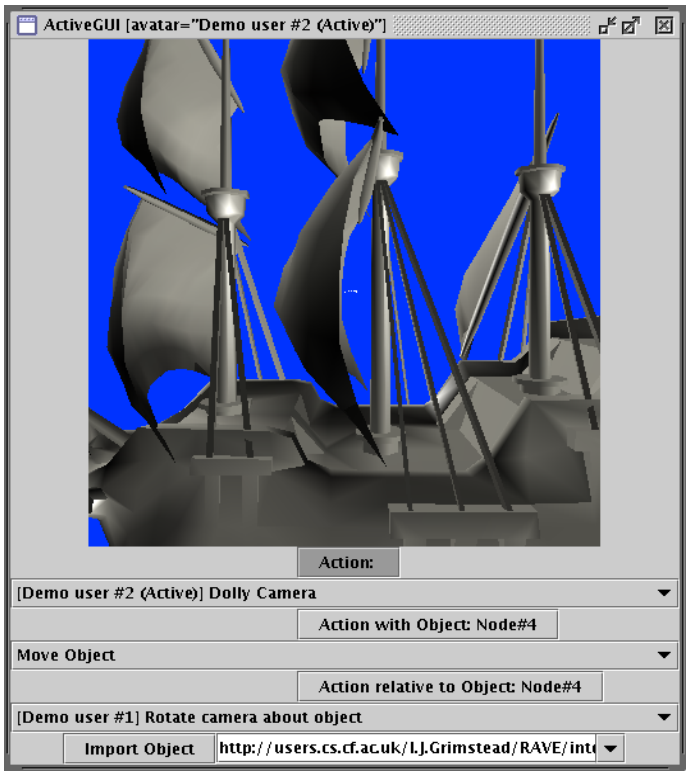

Figure 17: Zooming into the `galleon` Dataset

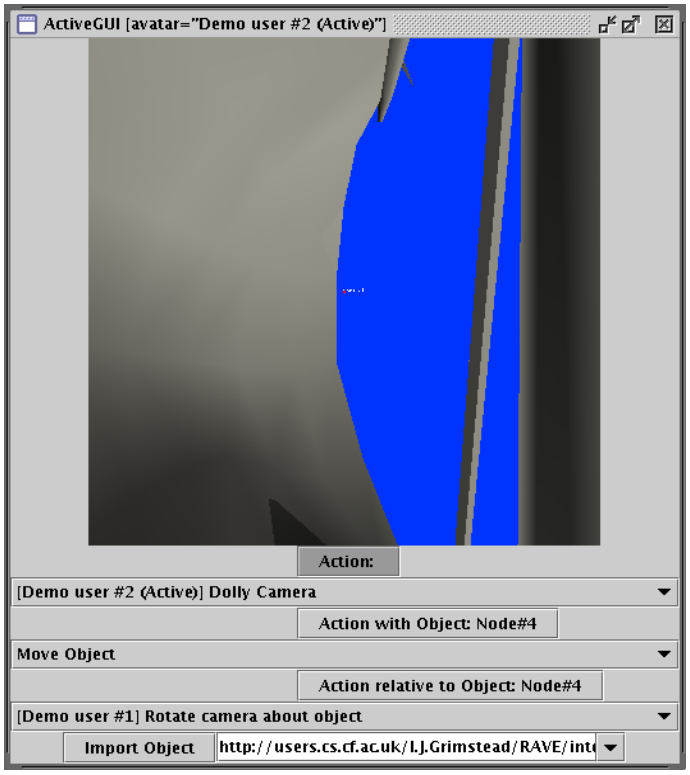

Figure 18: Zooming Further into the `galleon` Dataset

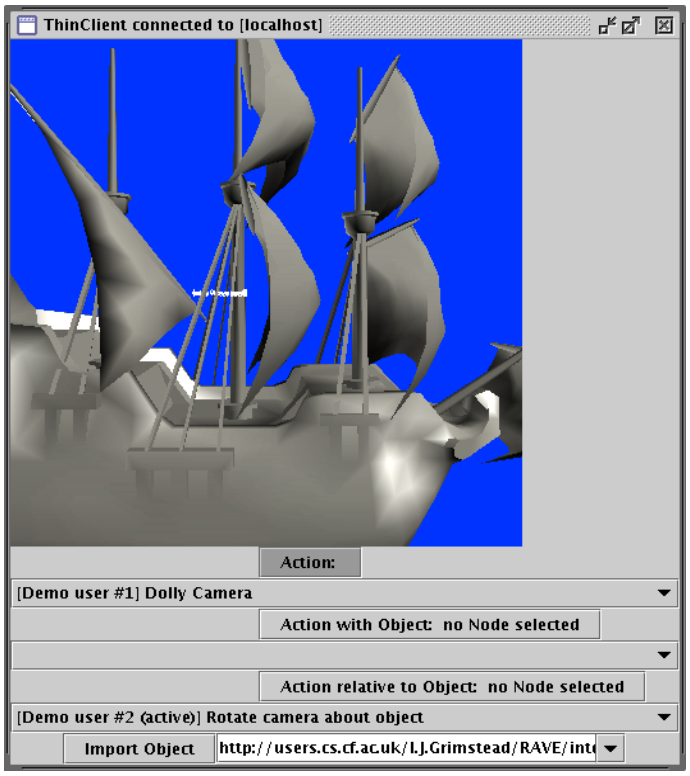
ThinClient connected to [localhost]
Action:
[Demo user #1] Dolly Camera
Action with Object: no Node selected
Action relative to Object: no Node selected
[Demo user #2 (active)] Rotate camera about object
Import Object
http://users.cs.cf.ac.uk/I.J.Grimstead/RAVE/int

Figure 19: RAVE Active Client—Initial View of the `galleon` Dataset

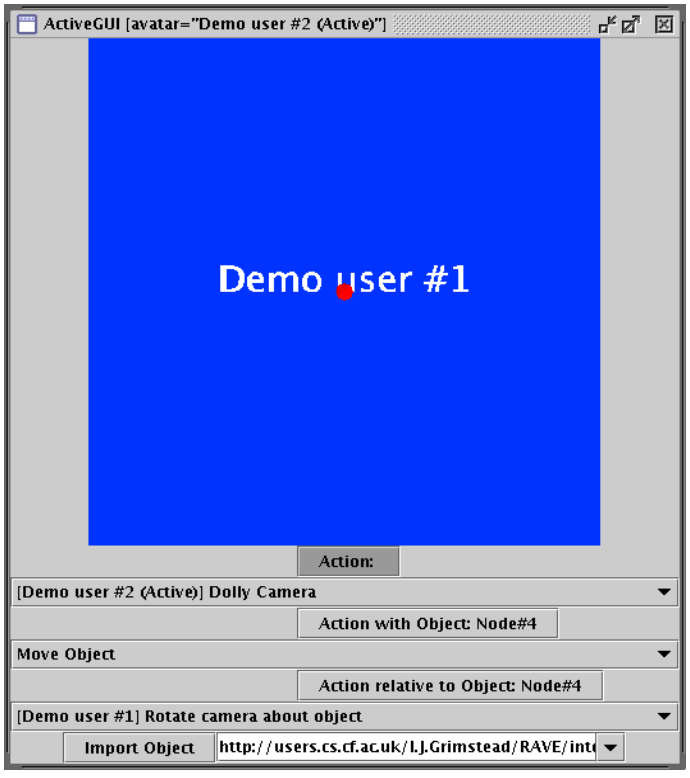

Figure 20: Zooming Further into the `galleon` Dataset

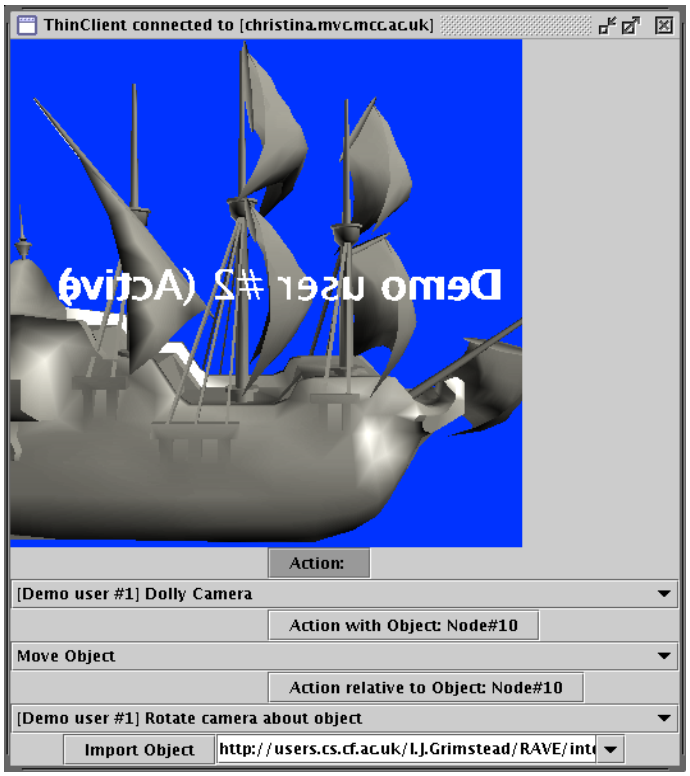

Figure 21: Zoom of Thin Client as Seen from Active Client

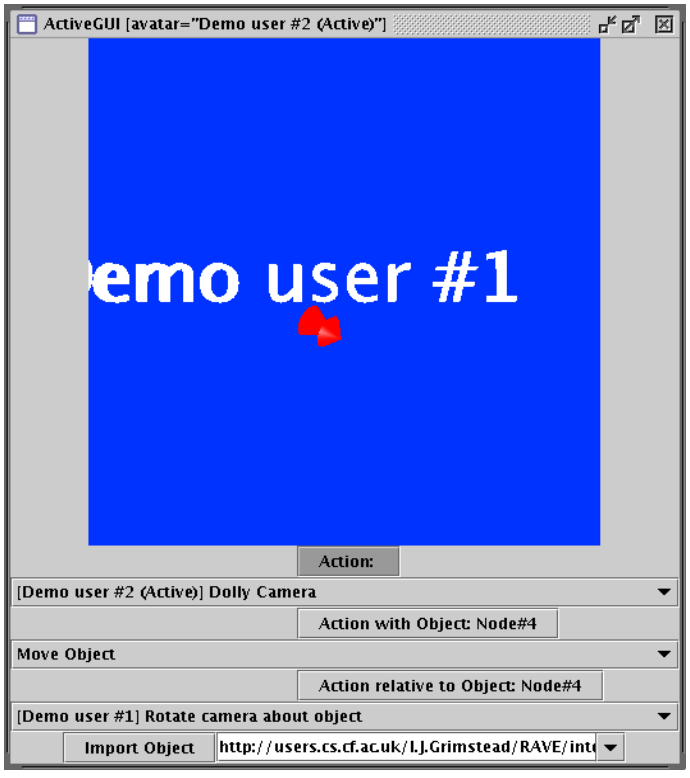

Figure 22: Zoom of Thin Client as Seen from Active Client

Figure 23: Photographs of the Zaurus C750

is replaced with a `Finish install` window. Click `OK`, close the `Package Installer` window, then close the `Add/Remove Software` window. A small window showing `Now updating information...` will appear, as the Zaurus Applications menu is updated.

Ensure the Zaurus is running in landscape orientation (with the keyboard exposed), and select the `Applications` tab. You will now see the RAVE Thin Client listed, compare with Figure 24. Our initial demonstration will be carried out in landscape mode rather than portrait—the differences will be explained later in a later section.

6.2 Starting the Thin Client

The RAVE Thin Client is started by simply tapping on the `RAVE Thin Client` icon in the `Applications` tab. This will produce the initial GUI, as in Figure 25. This GUI is much simpler than the RAVE Service Management GUI, so there are no drop-down menus or UDDI connectivity. At the time of writing, there was no publicly available UDDI toolkit for C/C++; only Java variants were available. Hence the GUI requires manual entry of any resource to be used.

For this test, we'll use `safleprofi.grid.cf.ac.uk` again. We already have a Render Service instance running, so we'll connect to that. Examining the `Create new Render Service instance` grouping box when creating our previous Thin Client (see Figure 9, the `Selected Service` URL for the Render Service is `http://safleprofi.grid.cf.ac.uk:28080/axis/services/RaveRender`. Type this URL into the `Render Service:` text box on the Zaurus. Similarly, enter "demo" for our `RS Instance`.

We already have created this Render Service instance, so we do not need to create it from this simplified GUI (so, do not press the `Create Render Service Instance` button). The information you have just entered is also used by the Thin Client to discover what Render Service to contact.

We now need to inform the GUI what size Thin Client we would like to use; we'll use a 400×400 image, so enter 400 in both the `Width:` and `Height:` boxes. Finally, we need to associate a user name with the Thin Client's avatar; for simplicity, we'll enter "PDA" in the `Username` box. We are now ready to launch the Thin Client; click on the `Connect to Render Service` button.

6.3 Using the Thin Client

An initial view of the `galleon` dataset on the Zaurus is presented in Figure 26. If you compare this to Figure 10, you will notice it is nearly identical, apart from the `Action` buttons and drop-down menus are placed to the right of the image.

Figure 24: Applications Pane on Zaurus Showing RAVE Thin Client

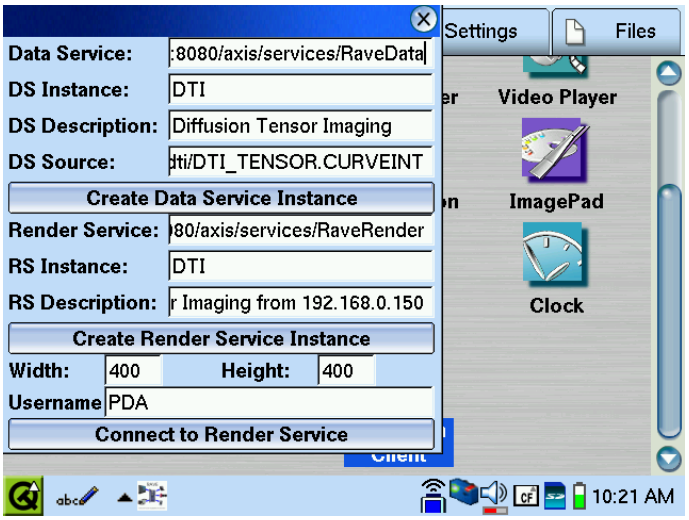

Figure 25: Initial Thin Client GUI on the Zaurus

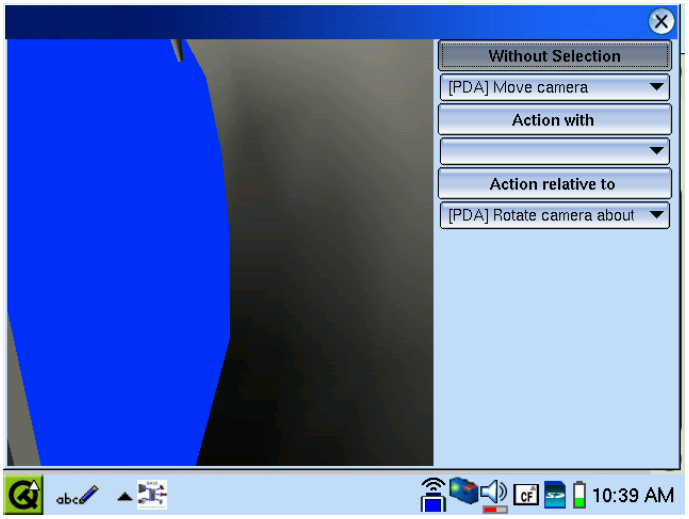

Figure 26: Initial View of the `galleon` Dataset on the Zaurus

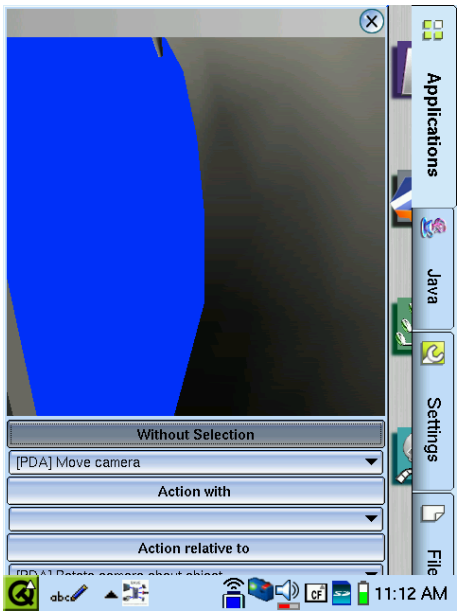

Figure 27: Portrait View of the `galleon` Dataset on the Zaurus

If you click on the `Action` drop-down menu, it is comparable to the previous menus (see Figure 11). All menus are generated from the same data, so whether you are running a Thin Client or an Active Client, or if the implementation was in Java or C/C++, the menus will remain constant.

You can now interact with the dataset in the same manner as with the Java implementations of the Thin and Active clients. Note that with the PDA, the slower network connection (11Mbit—which is shared with other users of a Wireless Access Point) will reveal the incremental, lossy image transmission technique.

6.4 Zaurus: Portrait or Landscape Mode?

The Zaurus we have used is a C750, which can rotate its screen (see Figure 23 for an image). We normally use the screen in landscape mode, as this reveals the keyboard (useful for entering URLs!). However, it can also function in portrait mode, looking more like a traditional PDA. If you wish to operate in portrait mode, you must first terminate the RAVE Thin Client by clicking on the small "×" on the top-right of the window.

Click on the `RAVE Thin Client` icon again to launch the client, and enter the details as before. Now, rotate the Zaurus to portrait mode. The GUI will rotate to match, so launch the new Thin Client. This time, the GUI will notice the orientation of the screen and change its on-screen layout accordingly, see Figure 27. The menus now appear underneath the received image, rather than to the right (otherwise the menus will be rendered off-screen with a 400×400 image and a 480×640 display).

7 RAVE Web Service Installation

RAVE has been tested using Jakarta-Tomcat [1] and Apache Axis [2]. Jakarta Tomcat is the web hosting application, acting as a container for Apache Axis. Apache Axis performs the marshalling of SOAP messages to present Web Services, whilst Jakarta Tomcat maintains the `http:` connection.

We will describe how to install Jakarta Tomcat, followed by Apache Axis and then generic RAVE installation.

7.1 Installing Jakarta Tomcat

Download the latest copy of Jakarta Tomcat from `http://jakarta.apache.org/site/downloads/downloads_tomcat-5.cgi`; select the latest stable build, currently tagged as version 5.5.7. Download `jakarta-tomcat-5.5.7.tar.gz`, marked as "5.5.7.tar.gz" on the web page. Create a directory where you wish to install it, and run `tar -xzf jakarta-tomcat-5.5.7.tar.gz`; this will create a sub-directory called `jakarta-tomcat-5.5.7`.

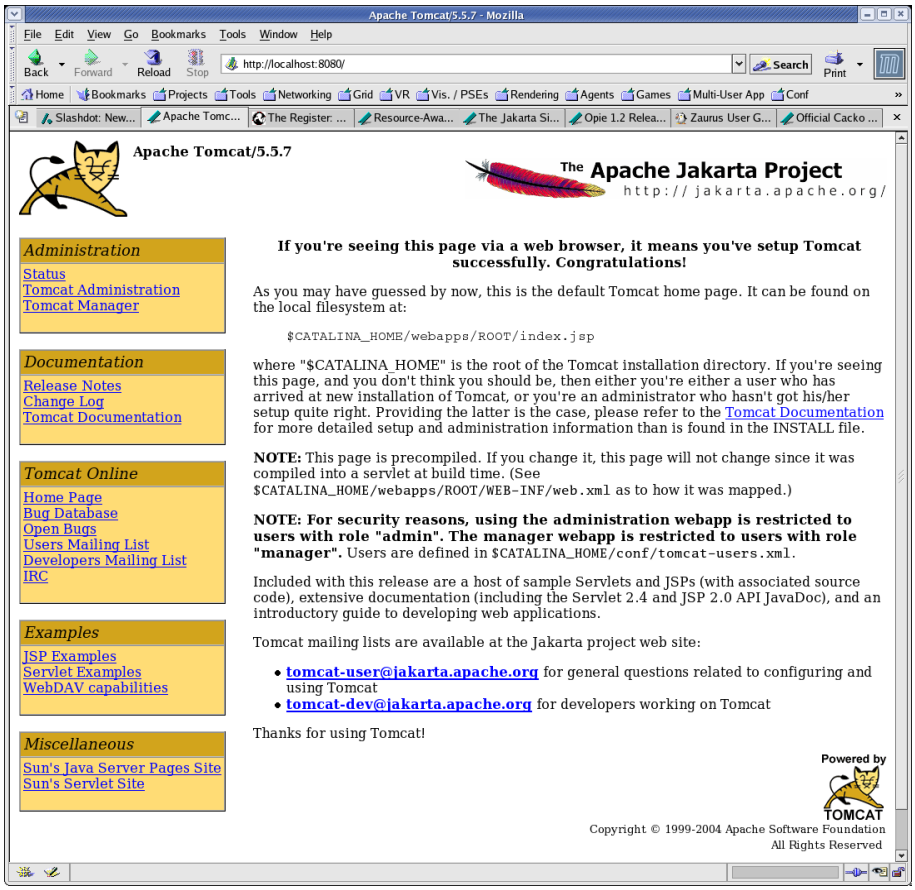
Apache Tomcat/5.5.7 - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://localhost:8080/
Search
Print
Apache Tomcat/5.5.7
The Apache Jakarta Project
http://jakarta.apache.org/
Administration
Status
Tomcat Administration
Tomcat Manager
Documentation
Release Notes
Change Log
Tomcat Documentation
Tomcat Online
Home Page
Bug Database
Open Bugs
Users Mailing List
Developers Mailing List
IRC
Examples
JSP Examples
Servlet Examples
WebDAV capabilities
Miscellaneous
Sun's Java Server Pages Site
Sun's Servlet Site
If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!
As you may have guessed by now, this is the default Tomcat home page. It can be found on the local filesystem at:
$CATALINA_HOME/webapps/ROOT/index.jsp
where "$CATALINA_HOME" is the root of the Tomcat installation directory. If you're seeing this page, and you don't think you should be, then either you're either a user who has arrived at new installation of Tomcat, or you're an administrator who hasn't got his/her setup quite right. Providing the latter is the case, please refer to the Tomcat Documentation for more detailed setup and administration information than is found in the INSTALL file.
NOTE: This page is precompiled. If you change it, this page will not change since it was compiled into a servlet at build time. (See $CATALINA_HOME/webapps/ROOT/WEB-INF/web.xml as to how it was mapped.)
NOTE: For security reasons, using the administration webapp is restricted to users with role "admin". The manager webapp is restricted to users with role "manager". Users are defined in $CATALINA_HOME/conf/tomcat-users.xml.
Included with this release are a host of sample Servlets and JSPs (with associated source code), extensive documentation (including the Servlet 2.4 and JSP 2.0 API JavaDoc), and an introductory guide to developing web applications.
Tomcat mailing lists are available at the Jakarta project web site:
tomcat-user@jakarta.apache.org for general questions related to configuring and using Tomcat
tomcat-dev@jakarta.apache.org for developers working on Tomcat
Thanks for using Tomcat!
Powered by
TOMCAT
Copyright © 1999-2004 Apache Software Foundation
All Rights Reserved

Figure 28: Initial View of Jakarta Tomcat's Root Web Page

Tomcat 5.5 is designed to run with Java 5.0 or later, so requires configuration to use JDK 1.4.*; we'll assume JDK 1.4 for this install. In this case we need the compatibility support as well; download `jakarta-tomcat-5.5.7-compat.tar.gz` (marked as "5.5.7 Compat tar.gz" on the web page). Extract the compatibility support if you're running in JDK 1.4.* by running `tar -xzf jakarta-tomcat-5.5.7-compat.tar.gz` in your created source directory. This will overwrite some files in the `jakarta-tomcat-5.5.7` subdirectory (namely the XML parser APIs).

To test your installation, move into the main jakarta directory and run `bin/start.sh`, viz:

```
cd <your root directory containing Jakarta Tomcat>
cd jakarta-tomcat-5.5.7
bin/startup.sh
```

This will start a web server running on port 8080; verify this by pointing a web browser at `http://localhost:8080/`. You should see a screen similar to Figure 28.

We now need to install Apache Axis to contain our Web Services.

7.2 Installing Apache Axis Inside Jakarta Tomcat

Apache Axis is hosted as part of the Apache Web Services project; download the latest release from `http://ws.apache.org/axis/releases.html` (at the time of writing, 1.2 release candidate 3 was available).

Extract the archive into your base directory (we used `jakarta-tomcat`), not the actual Jakarta Tomcat installation directory; we will extract parts of the Apache Axis and insert these directly into the Jakarta Tomcat installation.

Initially, copy the `webapps` subdirectory from Apache Axis into Jakarta Tomcat:

```
cd <your root directory containing Apache Axis and Jakarta Tomcat>
cp -r axis-1\_2RC3/webapps jakarta-tomcat-5.5.7
```

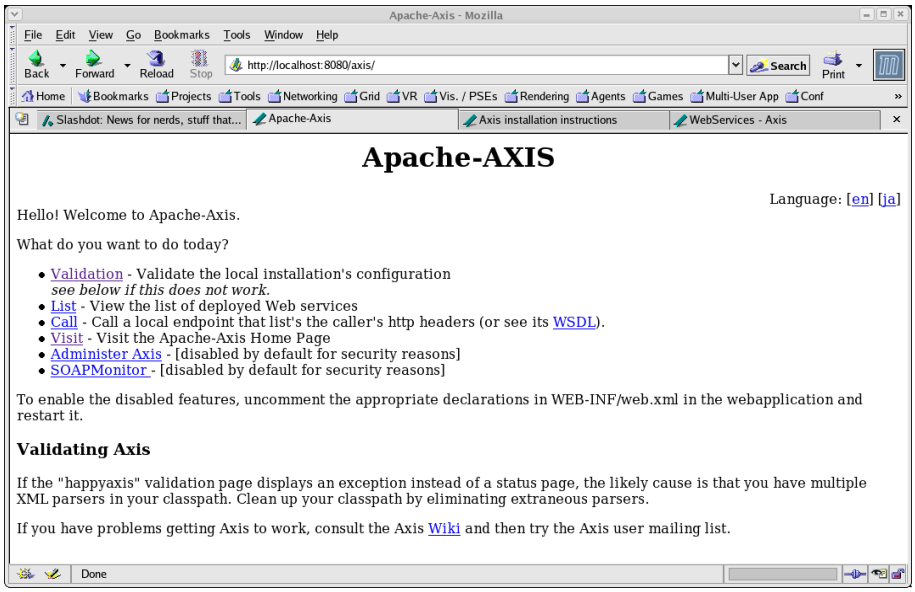
Apache-Axis - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://localhost:8080/axis/
Search
Print
Home Bookmarks Projects Tools Networking Grid VR Vis. / PSEs Rendering Agents Games Multi-User App Conf
Slashdot: News for nerds, stuff that...
Apache-Axis
Axis installation instructions
WebServices - Axis
Apache-AXIS
Language: [en] [ja]
Hello! Welcome to Apache-Axis.
What do you want to do today?
Validation - Validate the local installation's configuration
see below if this does not work.
List - View the list of deployed Web services
Call - Call a local endpoint that list's the caller's http headers (or see its WSDL).
Visit - Visit the Apache-Axis Home Page
Administer Axis - [disabled by default for security reasons]
SOAPMonitor - [disabled by default for security reasons]
To enable the disabled features, uncomment the appropriate declarations in WEB-INF/web.xml in the webapplication and restart it.
Validating Axis
If the "happyaxis" validation page displays an exception instead of a status page, the likely cause is that you have multiple XML parsers in your classpath. Clean up your classpath by eliminating extraneous parsers.
If you have problems getting Axis to work, consult the Axis Wiki and then try the Axis user mailing list.
Done

Figure 29: Initial View of Apache Axis' Root Web Page

Now direct your web browser at `http://localhost:8080/axis` (this will display the Apache Axis status on your machine). Your web browser will return a page akin to Figure 29.

Click on the `Validation` link—this will verify if Apache Axis is working properly. You should see output similar to the following:

```
Axis Happiness Page
Examining webapp configuration

Language: [en] [ja]
Needed Components

    * Found SAAJ API ( javax.xml.soap.SOAPMessage ) at
    /home/user/jakarta-tomcat/jakarta-tomcat-5.5.7/webapps/axis/WEB-INF/lib/saaj.jar

    * Found JAX-RPC API ( javax.xml.rpc.Service ) at
    /home/user/jakarta-tomcat/jakarta-tomcat-5.5.7/webapps/axis/WEB-INF/lib/jaxrpc.jar

    * Found Apache-Axis ( org.apache.axis.transport.http.AxisServlet ) at
    /home/user/jakarta-tomcat/jakarta-tomcat-5.5.7/webapps/axis/WEB-INF/lib/axis.jar

    * Found Jakarta-Commons Discovery ( org.apache.commons.discovery.Resource ) at
    /home/user/jakarta-tomcat/jakarta-tomcat-5.5.7/webapps/axis/WEB-INF/lib/commons-discovery.jar

    * Found Jakarta-Commons Logging ( org.apache.commons.logging.Log ) at
    /home/user/jakarta-tomcat/jakarta-tomcat-5.5.7/bin/commons-logging-api.jar

    * Found Log4j ( org.apache.log4j.Layout ) at
    /home/user/jakarta-tomcat/jakarta-tomcat-5.5.7/webapps/axis/WEB-INF/lib/log4j-1.2.8.jar

    * Found IBM's WSDL4Java ( com.ibm.wsdl.factory.WSDLFactoryImpl ) at
    /home/user/jakarta-tomcat/jakarta-tomcat-5.5.7/webapps/axis/WEB-INF/lib/wsdl4j.jar

    * Found JAXP implementation ( javax.xml.parsers.SAXParserFactory ) at an unknown location

      Error: could not find class javax.activation.DataHandler from file activation.jar
      Axis will not work.
```

```
      See http://java.sun.com/products/javabeans/glasgow/jaf.html

Optional Components

      Warning: could not find class javax.mail.internet.MimeMessage from file mail.jar
      Attachments will not work.
      See http://java.sun.com/products/javamail/

      Warning: could not find class org.apache.xml.security.Init from file xmlsec.jar
      XML Security is not supported.
      See http://xml.apache.org/security/

    * Found Java Secure Socket Extension ( javax.net.ssl.SSLSocketFactory ) at an unknown location

1 core axis library(ies) are missing2 wanted optional axis librar(ies) are missing
```

We are missing a library, but helpfully Apache Axis points us to the URL http://java.sun.com/products/javabeans/glasgow/jaf.html. This is the JavaBeans Activation Framework, which is used by Apache Axis. From this page you can download the latest framework (version 1.0.2) as a `.zip` file. Unzip this into your root installation directory (where Jakarta Tomcat and Apache Axis are subdirectories of this). This will create a new directory called `jaf-1.0.2`. Copy the `activation.jar` file into Jakarta Tomcat for Apache Axis to use, and restart Jakarta Tomcat:

```
cd <your root directory containing Apache Axis and Jakarta Tomcat>
cp -r jaf-1.0.2/activation.jar jakarta-tomcat-5.5.7/webapps/axis/WEB-INF/lib
cd jakarta-tomcat-5.5.7
bin/shutdown.sh
bin/startup.sh
```

Go back to your web browser and reload the `Validation` page—this will verify if Apache Axis is working properly. You should now see output similar to the following:

```
Axis Happiness Page
Examining webapp configuration

Language: [en] [ja]
Needed Components

    * Found SAAJ API ( javax.xml.soap.SOAPMessage ) at
    /home/user/jakarta-tomcat/jakarta-tomcat-5.5.7/webapps/axis/WEB-INF/lib/saaj.jar

    * Found JAX-RPC API ( javax.xml.rpc.Service ) at
    /home/user/jakarta-tomcat/jakarta-tomcat-5.5.7/webapps/axis/WEB-INF/lib/jaxrpc.jar

    * Found Apache-Axis ( org.apache.axis.transport.http.AxisServlet ) at
    /home/user/jakarta-tomcat/jakarta-tomcat-5.5.7/webapps/axis/WEB-INF/lib/axis.jar

    * Found Jakarta-Commons Discovery ( org.apache.commons.discovery.Resource ) at
    /home/user/jakarta-tomcat/jakarta-tomcat-5.5.7/webapps/axis/WEB-INF/lib/commons-discovery.jar

    * Found Jakarta-Commons Logging ( org.apache.commons.logging.Log ) at
    /home/user/jakarta-tomcat/jakarta-tomcat-5.5.7/bin/commons-logging-api.jar

    * Found Log4j ( org.apache.log4j.Layout ) at
    /home/user/jakarta-tomcat/jakarta-tomcat-5.5.7/webapps/axis/WEB-INF/lib/log4j-1.2.8.jar

    * Found IBM's WSDL4Java ( com.ibm.wsdl.factory.WSDLFactoryImpl ) at
    /home/user/jakarta-tomcat/jakarta-tomcat-5.5.7/webapps/axis/WEB-INF/lib/wsdl4j.jar
```

```
    * Found JAXP implementation ( javax.xml.parsers.SAXParserFactory ) at an unknown location


    * Found Activation API ( javax.activation.DataHandler ) at
    /home/user/jakarta-tomcat/jakarta-tomcat-5.5.7/webapps/axis/WEB-INF/lib/activation.jar

Optional Components

      Warning: could not find class javax.mail.internet.MimeMessage from file mail.jar
      Attachments will not work.
      See http://java.sun.com/products/javamail/

      Warning: could not find class org.apache.xml.security.Init from file xmlsec.jar
      XML Security is not supported.
      See http://xml.apache.org/security/

    * Found Java Secure Socket Extension ( javax.net.ssl.SSLSocketFactory ) at an unknown location

The core axis libraries are present.2 wanted optional axis librar(ies) are missing
```

The only missing items are email and XML security related, which we do not use. Hence Apache Axis is now ready to have RAVE installed.

7.3 Installing the RAVE Web Service

We need to copy the support libraries into Jakarta Tomcat, as RAVE uses third party libraries to enable support for foreign file formats. We use:

Java3D Rendering technology that makes use of OpenGL

Xj3D The Java implementation of X3D, the replacement of VRML with XML format files

Note that Jakarta Tomcat is restricted in not permitting calls to Java methods named `java.*` or `javax.*`, unless the calls are made to jar files inside the "endorsed" directory. Hence some of the libraries we rely on must be installed here, rather than inside the Apache Axis directory of Jakarta Tomcat.

7.3.1 Java3D Installation

Java3D must be installed in the local Java Virtual Machine, as it contains native code (run-time libraries) that must be installed there. However, it contains classes named `java.*`, so we must copy the jar files into the endorsed directory of Jakarta Tomcat:

```
cd <your root directory containing Apache Axis and Jakarta Tomcat>
cp $JAVA_HOME/jre/lib/ext/j3d*.jar jakarta-tomcat-5.5.7/common/endorsed
```

7.3.2 Xj3D Installation

Download the latest copy of Xj3D from http://www.xj3d.org/snapshots.html (the latest at the time of writing was M10). Currently, Xj3D is packaged as a single jar file which self-extracts itself. Download the release (for instance, http://www.xj3d.org/downloads/Xj3D-M10-linuxx86.jar) into your root directory, and run:

```
java -jar Xj3D-M10-linuxx86.jar
```

This will pop-up an installation window akin to Figure 30. Click on the `Next` button, accept the license, click `Next` and then enter a directory to install the files. We selected the `Xj3D` subdirectory of the root of our installation (where Apache Axis, JAF, etc. were downloaded). Select which pack to install (there is only one—the "base", which is mandatory), then select if you wish for an on-screen shortcut, then start the installation (clicking `Next` as appropriate). Once complete, click `Next` and then `Done`. Xj3D is now installed, so we can copy the appropriate jar files into Jakarta Tomcat/Apache Axis. The required jar files are:

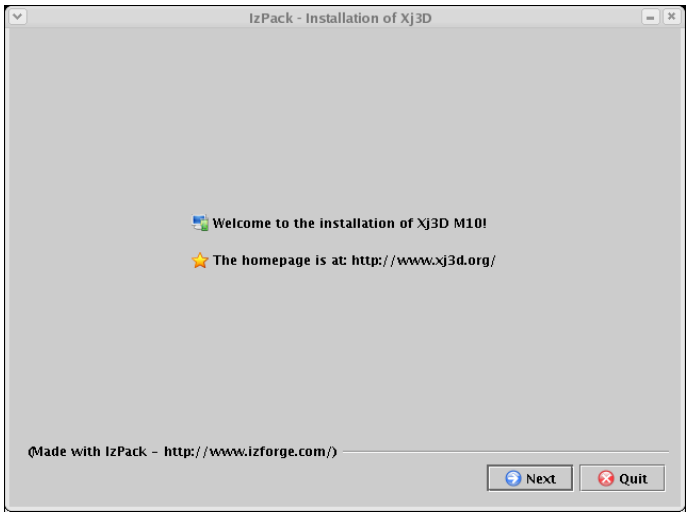
IzPack - Installation of Xj3D
Welcome to the installation of Xj3D M10!
The homepage is at: http://www.xj3d.org/
(Made with IzPack - http://www.izforge.com/)
Next
Quit

Figure 30: The Xj3D Installer

- j3d-org.jar
- uri.jar
- vlc_uri.jar
- xj3d-common.jar
- xj3d-core.jar
- xj3d-ecmascript.jar
- xj3d-j3d.jar
- xj3d-jsai.jar
- xj3d-parser.jar
- xj3d-render.jar
- xj3d-runtime.jar
- xj3d-sav.jar
- xj3d-script-base.jar

Copy these files into the common/endorsed directory, viz:

```
cd <your root directory containing Apache Axis and Jakarta Tomcat>
cp Xj3D/jars/j3d-org.jar jakarta-tomcat-5.5.7/common/endorsed/
cp Xj3D/jars/uri.jar jakarta-tomcat-5.5.7/common/endorsed/
cp Xj3D/jars/vlc_uri.jar jakarta-tomcat-5.5.7/common/endorsed/
cp Xj3D/jars/xj3d-common.jar jakarta-tomcat-5.5.7/common/endorsed/
cp Xj3D/jars/xj3d-core.jar jakarta-tomcat-5.5.7/common/endorsed/
cp Xj3D/jars/xj3d-ecmascript.jar jakarta-tomcat-5.5.7/common/endorsed/
cp Xj3D/jars/xj3d-j3d.jar jakarta-tomcat-5.5.7/common/endorsed/
cp Xj3D/jars/xj3d-jsai.jar jakarta-tomcat-5.5.7/common/endorsed/
cp Xj3D/jars/xj3d-parser.jar jakarta-tomcat-5.5.7/common/endorsed/
cp Xj3D/jars/xj3d-render.jar jakarta-tomcat-5.5.7/common/endorsed/
cp Xj3D/jars/xj3d-runtime.jar jakarta-tomcat-5.5.7/common/endorsed/
cp Xj3D/jars/xj3d-sav.jar jakarta-tomcat-5.5.7/common/endorsed/
cp Xj3D/jars/xj3d-script-base.jar jakarta-tomcat-5.5.7/common/endorsed/
```

7.3.3 RAVE Jar File Installation

Copy the `RAVE-0.4-alpha.jar` file into the Apache Axis area of Jakarta Tomcat:

```
cd <your root directory containing Apache Axis and Jakarta Tomcat>
cp RAVE-0.4-alpha.jar jakarta-tomcat-5.5.7/webapps/axis/WEB-INF/lib
```

Note that now we have updated jar files inside Jakarta Tomcat, it must be restarted to ensure the new jar files are detected in the classpath, etc. A simple approach is to shut down and restart Jakarta Tomcat:

```
cd <your root directory containing Apache Axis and Jakarta Tomcat>
cd jakarta-tomcat-5.5.7
bin/shutdown.sh
bin/startup.sh
```

We now have to inform Apache Axis that new services have been installed; this is discussed in the next section.

8 Data Service Installation

We first inform the Apache Axis installation that a new service is available, and (optionally) register the new service on the UDDI server.

8.1 Informing Apache Axis of the RAVE Data Service

We will announce RAVE to Apache Axis; copy the `ws-deploy-data.wsdd` file to the local directory, and run:

```
java -cp axis.jar:commons-discovery.jar:commons-logging.jar:jaxrpc.jar\
:log4j-1.2.8.jar:saaj.jar:uddi4j.jar:xercesImpl.jar:xmlParserAPIs.jar\
:RAVE-0.4-alpha.jar org.apache.axis.client.AdminClient \
-lhttp://localhost:8080/axis/servlet/AxisServlet ws-deploy-data.wsdd
```

This assumes you're using the localhost, and that you are still using port 8080 for Jakarta Tomcat/Apache Axis. You should see output similar to the following:

```
Processing file ws-deploy-data.wsdd
<Admin>Done processing</Admin>
```

Point your web browser at the URL http://localhost:8080/axis/servlet/AxisServlet (this will list currently available services). You should see output similar to:

```
And now... Some Services
    * AdminService (wsdl)
          o AdminService
    * Version (wsdl)
          o getVersion
    * RaveData (wsdl)
          o shutdown
          o getVersion
          o getDescription
          o getBuildTag
          o setEndPoint
          o requestSocketMHLog
          o setSocketMHLogLevel
          o getNumInstances
          o getTotalNumSubscribers
          o getNumSubscribers
          o getInstanceName
```

```
    o instanceExists
    o createInstance
    o ping
    o setPortRange
    o subscribe
    o getBootstrapPortNum
    o getUpdatePortNum
    o addGeometry
    o addDefaultViewpoint
    o saveWorld
    o getNumberOfTriangles
    o getMatricesPerSec
    o getSystemMemoryFree
    o getNetworkTimingSocket
    o ensureLit
    o getMemoryInUseGuesstimate
```

Note the "RaveData" service being listed (and its related service entry points); this is our RAVE Data Service.

8.2 Informing the WeSC UDDI Server of the RAVE Data Service

If you wish to declare this resource on the WeSC UDDI server, run:

```
java -cp axis.jar:commons-discovery.jar:commons-logging.jar:jaxrpc.jar\
:log4j-1.2.8.jar:saaj.jar:uddi4j.jar:xercesImpl.jar:xmlParserAPIs.jar\
:RAVE-0.4-alpha.jar \
uk.ac.cardiff.cs.rave.net.ws.data.utils.DataServiceDeployToUDDI
```

This will produce output akin to the following (example is from when `safleprofi.grid.cf.ac.uk` was initialised; note that this machine uses port 28080, you should see 8080 in your output):

```
- Using UDDI inquiry URL=[http://uddi.wesc.ac.uk:8334/juddi/inquiry]
- Using UDDI publish URL=[http://uddi.wesc.ac.uk:8334/juddi/publish]
RAVE Business ID: 7EDF6D20-1222-11D9-B11C-8E608C4352A8
RAVE RAVE Data Service: Tech Model ID: uuid:7F2E0250-1222-11D9-B11C-9B79FB8F86F0
Reading RAVE services from UDDI: Time taken to scan and receive services: 1.393s
RAVE RAVE Data Service: Service ID: 7F8C00D0-1222-11D9-B11C-82176714E2B7
Checking for service access point: http://safleprofi.grid.cf.ac.uk:28080/axis/services/RaveData
Deploying access point [http://safleprofi.grid.cf.ac.uk:28080/axis/services/RaveData]
description="RAVE Data Web Service [published on 2005-03-03 17:08:40]"
Service access point not present; publishing new entry
Published service at [http://safleprofi.grid.cf.ac.uk:28080/axis/services/RaveData]
```

If you now re-run the RAVE Service Management GUI, you will see your service listed under the `Available Data Services`.

8.3 Removing the RAVE Data Service from Apache Axis

If you wish to no longer provide a Data Service, then you must inform Apache Axis. Copy the `ws-undeploy-data.wsdd` file to the local directory, and run:

```
java -cp axis.jar:commons-discovery.jar:commons-logging.jar:jaxrpc.jar\
:log4j-1.2.8.jar:saaj.jar:uddi4j.jar:xercesImpl.jar:xmlParserAPIs.jar\
:RAVE-0.4-alpha.jar org.apache.axis.client.AdminClient \
-lhttp://localhost:8080/axis/servlet/AxisServlet ws-undeploy-data.wsdd
```

This assumes you're using the localhost, and that you are still using port 8080 for Jakarta Tomcat/Apache Axis. You should see output similar to the following:

```
Processing file ws-undeploy-data.wsdd
<Admin>Done processing</Admin>
```

Point your web browser at the URL http://localhost:8080/axis/servlet/AxisServlet (this will list currently available services). You will notice that the entry related to `RaveData` has been removed.

8.4 Removing the RAVE Data Service from the WeSC UDDI Server

If you wish to declare this resource as no longer available on the WeSC UDDI server, then we must undeploy it there as well; run:

```
java -cp axis.jar:commons-discovery.jar:commons-logging.jar:jaxrpc.jar\
:log4j-1.2.8.jar:saaj.jar:uddi4j.jar:xercesImpl.jar:xmlParserAPIs.jar\
:RAVE-0.4-alpha.jar \
uk.ac.cardiff.cs.rave.net.ws.data.utils.DataServiceUndeployToUDDI
```

This will produce output akin to the following (example is from when `safleprofi.grid.cf.ac.uk` was initialised; note that this machine uses port 28080, you should see 8080 in your output):

```
- Using UDDI inquiry URL=[http://uddi.wesc.ac.uk:8334/juddi/inquiry]
- Using UDDI publish URL=[http://uddi.wesc.ac.uk:8334/juddi/publish]
RAVE Business ID: 7EDF6D20-1222-11D9-B11C-8E608C4352A8
RAVE RAVE Data Service: Tech Model ID: uuid:7F2E0250-1222-11D9-B11C-9B79FB8F86F0
RAVE RAVE Data Service: Service ID: 7F8C00D0-1222-11D9-B11C-82176714E2B7
Checking for service access point: http://safleprofi.grid.cf.ac.uk:28080/axis/services/RaveData
Undeploying access point [http://safleprofi.grid.cf.ac.uk:28080/axis/services/RaveData]
Undeploying Service binding [http://safleprofi.grid.cf.ac.uk:28080/axis/services/RaveData]
```

If you now re-run the RAVE Service Management GUI, you will see your service is no longer listed under the `Available Data Services`.

9 Render Service Installation

This consists of a similar sequence of steps as that of the Data Service Installation, only this time our scripts will refer to the Render Service instead.

9.1 Note on User Access to Graphics Hardware

Note that the user running Apache Axis will now need access to the graphics hardware (as RAVE uses it, and will be running inside Apache Axis). You may need to adjust security settings to enable this on your machine.

9.1.1 Linux Specifics

The first area to check is the security settings on the actual graphics device; they may default to be owned by `root` and group `root`, with permissions 600. This means that only user `root` can access the hardware, unless the user is logged onto the console. Change the access permissions to 666, but note that the PAM settings may also need to be modified (this is true with NVidia cards; check the readme that comes with NVidia's hardware accelerated driver).

Secondly, the user running Jakarta Tomcat must have access to the X11 session. The simplest way is to log in to the console as that user, or to run:

```
xhost +local:
```

on the command line (this will enable all locally-logged in users to access the X11 screen). X11 access is needed, as this is how OpenGL contacts the graphics hardware; OpenGL is the underlying rendering mechanism used by Java3D, which is in turn used by RAVE.

9.2 Informing Apache Axis of the RAVE Render Service

We will announce RAVE to Apache Axis; copy the `ws-deploy-render.wsdd` file to the local directory, and run:

```
java -cp axis.jar:commons-discovery.jar:commons-logging.jar:jaxrpc.jar\
:log4j-1.2.8.jar:saaj.jar:uddi4j.jar:xercesImpl.jar:xmlParserAPIs.jar\
:RAVE-0.4-alpha.jar org.apache.axis.client.AdminClient \
-lhttp://localhost:8080/axis/servlet/AxisServlet ws-deploy-render.wsdd
```

This assumes you're using the localhost, and that you are still using port 8080 for Jakarta Tomcat/Apache Axis. You should see output similar to the following:

```
Processing file ws-deploy-render.wsdd
<Admin>Done processing</Admin>
```

Point your web browser at the URL http://localhost:8080/axis/servlet/AxisServlet (this will list currently available services). You should see output similar to before, with the exception of the additional lines before the `RaveData` section:

```
# RaveRender (wsdl)
    * shutdown
    * getVersion
    * getDescription
    * getBuildTag
    * setEndPoint
    * requestSocketMHLog
    * setSocketMHLogLevel
    * getNumInstances
    * getTotalNumSubscribers
    * getNumSubscribers
    * getInstanceName
    * instanceExists
    * instanceIsInitialising
    * createInstance
    * ping
    * setPortRange
    * subscribe
    * unsubscribe
    * spawnLocalViewer
    * getPortNum
    * startRenderStream
    * stopRenderStream
    * clientPosAdd
    * clientPosSet
    * clientMatrixSet
    * clientRollAdd
    * getMatricesPerSec
    * getPolygonsPerSec
    * getSystemMemoryFree
    * getNetworkTimingSocket
    * selectObject
    * areAllInteractionsContinuous
    * getAllInteractions
    * areAllInteractionsWhenSelectedForNodeContinuous
    * getAllInteractionsWhenSelectedForNode
    * areAllInteractionsWithObjectContinous
    * getAllInteractionsWithObject
    * doInteraction
    * doInteractionLogged
```

```
* doInteractionWhenSelected
* doInteractionWhenSelectedLogged
* doInteractionWithObject
* doInteractionWithObjectLogged
* getConnectedDataServiceUrl
* getConnectedDataServiceInstance
* requestTiledAssistance
* getMemoryInUseGuesstimate
* getNumberOfTriangles
* loadObjectFromUrl
```

9.3 Informing the WeSC UDDI Server of the RAVE Render Service

If you wish to declare this resource on the WeSC UDDI server, run:

```
java -cp axis.jar:commons-discovery.jar:commons-logging.jar:jaxrpc.jar\
:log4j-1.2.8.jar:saaj.jar:uddi4j.jar:xercesImpl.jar:xmlParserAPIs.jar\
:RAVE-0.4-alpha.jar \
uk.ac.cardiff.cs.rave.net.ws.render.utils.RenderServiceDeployToUDDI
```

This will produce output akin to the following (example is from when `safleprofi.grid.cf.ac.uk` was initialised; note that this machine uses port 28080, you should see 8080 in your output):

```
- Using UDDI inquiry URL=[http://uddi.wesc.ac.uk:8334/juddi/inquiry]
- Using UDDI publish URL=[http://uddi.wesc.ac.uk:8334/juddi/publish]
RAVE Business ID: 7EDF6D20-1222-11D9-B11C-8E608C4352A8
RAVE RAVE Render Service: Tech Model ID: uuid:8EA2EB10-1222-11D9-B11C-9C99BA18CB58
Reading RAVE services from UDDI: Time taken to scan and receive services: 1.266s
RAVE RAVE Render Service: Service ID: 90F51870-1222-11D9-B11C-D1A962B40FFC
Checking for service access point: http://safleprofi.grid.cf.ac.uk:28080/axis/services/RaveRender
Deploying access point [http://safleprofi.grid.cf.ac.uk:28080/axis/services/RaveRender]
description="RAVE Render Web Service [published on 2005-03-03 17:28:14]"
Service access point not present; publishing new entry
Published service at [http://safleprofi.grid.cf.ac.uk:28080/axis/services/RaveRender]
```

If you now re-run the RAVE Service Management GUI, you will see your service listed under the `Available Render Services`.

9.4 Removing the RAVE Render Service from Apache Axis

If you wish to no longer provide a Render Service, then you must inform Apache Axis. Copy the `ws-undeploy-render.wsdd` file to the local directory, and run:

```
java -cp axis.jar:commons-discovery.jar:commons-logging.jar:jaxrpc.jar\
:log4j-1.2.8.jar:saaj.jar:uddi4j.jar:xercesImpl.jar:xmlParserAPIs.jar\
:RAVE-0.4-alpha.jar org.apache.axis.client.AdminClient \
-lhttp://localhost:8080/axis/servlet/AxisServlet ws-undeploy-data.wsdd
```

This assumes you're using the localhost, and that you are still using port 8080 for Jakarta Tomcat/Apache Axis. You should see output similar to the following:

```
Processing file ws-undeploy-render.wsdd
<Admin>Done processing</Admin>
```

Point your web browser at the URL http://localhost:8080/axis/servlet/AxisServlet (this will list currently available services). You will notice that the entry related to `RaveRender` has been removed.

9.5 Removing the RAVE Render Service from the WeSC UDDI Server

If you wish to declare this resource as no longer available on the WeSC UDDI server, then we must undeploy it there as well; run:

```
java -cp axis.jar:commons-discovery.jar:commons-logging.jar:jaxrpc.jar\
:log4j-1.2.8.jar:saaj.jar:uddi4j.jar:xercesImpl.jar:xmlParserAPIs.jar\
:RAVE-0.4-alpha.jar \
uk.ac.cardiff.cs.rave.net.ws.render.utils.RenderServiceUndeployToUDDI
```

This will produce output akin to the following (example is from when `safleprofi.grid.cf.ac.uk` was initialised; note that this machine uses port 28080, you should see 8080 in your output):

```
- Using UDDI inquiry URL=[http://uddi.wesc.ac.uk:8334/juddi/inquiry]
- Using UDDI publish URL=[http://uddi.wesc.ac.uk:8334/juddi/publish]
RAVE Business ID: 7EDF6D20-1222-11D9-B11C-8E608C4352A8
RAVE RAVE Render Service: Tech Model ID: uuid:8EA2EB10-1222-11D9-B11C-9C99BA18CB58
RAVE RAVE Render Service: Service ID: 90F51870-1222-11D9-B11C-D1A962B40FFC
Checking for service access point: http://safleprofi.grid.cf.ac.uk:28080/axis/services/RaveRender
Undeploying access point [http://safleprofi.grid.cf.ac.uk:28080/axis/services/RaveRender]
Undeploying Service binding [http://safleprofi.grid.cf.ac.uk:28080/axis/services/RaveRender]
```

If you now re-run the RAVE Service Management GUI, you will see your service is no longer listed under the `Available Render Services`.

Note that if you run the undeployment twice, it will check if the service exists before trying to undeploy it, producing output similar to:

```
- Using UDDI inquiry URL=[http://uddi.wesc.ac.uk:8334/juddi/inquiry]
- Using UDDI publish URL=[http://uddi.wesc.ac.uk:8334/juddi/publish]
RAVE Business ID: 7EDF6D20-1222-11D9-B11C-8E608C4352A8
RAVE RAVE Render Service: Tech Model ID: uuid:8EA2EB10-1222-11D9-B11C-9C99BA18CB58
RAVE RAVE Render Service: Service ID: 90F51870-1222-11D9-B11C-D1A962B40FFC
Checking for service access point: http://safleprofi.grid.cf.ac.uk:28080/axis/services/RaveRender
```

10 Errata

In this section we list the outstanding support items in RAVE.

10.1 Removal of Client on Exit

At present, when a client exists, it leaves its avatar in the system. We will eventually remove avatars as their owning clients exit the RAVE session.

10.2 Session Shutdown

Not yet implemented; to clean a service host, you will need to restart Tomcat/Apache Axis. Eventually we will add a method into the GUI to support this.

11 Revision Log

The CVS revision log is now presented, to reveal changes made to this Chapter.

```
$Log: Guide-InstallationAndUser.tex,v $
Revision 1.14  2005/03/09 13:03:58  scmijg
Added note about X11 access under Linux...

Revision 1.13  2005/03/04 19:17:24  scmijg
Spellchecked...
```

```
Revision 1.12  2005/03/04 16:00:20  scmijg
Finished installation and user guide.
Can also produce stand-alone install/user guide.

Revision 1.11  2005/03/04 15:35:54  scmijg
Install/uninstall completed.

Revision 1.10  2005/03/04 15:18:20  scmijg
Rave/Data services added.
Working on undeployment...

Revision 1.9  2005/03/03 17:10:02  scmijg
Data service deployment...

Revision 1.8  2005/03/03 14:17:38  scmijg
Tidied verbatim bits...

Revision 1.7  2005/03/03 13:33:30  scmijg
Working on Tomcat/Axis installation...

Revision 1.6  2005/03/03 12:09:49  scmijg
Thin clients complete (inc. Zaurus). Maybe.

Onto Tomcat...

Revision 1.5  2005/03/02 23:02:18  scmijg
Active Client nearly finished.

Revision 1.4  2005/03/02 10:06:35  scmijg
Moved javadoc files to javadoc/*
Corrected Part-Implementation.tex to match
Working on ActiveClient guide...

Revision 1.3  2005/03/02 08:38:01  scmijg
Slowly working on Active Client...

Revision 1.2  2005/03/01 17:08:43  scmijg
Touched on Active client...

Revision 1.1  2005/03/01 16:44:13  scmijg
Merged Installation and User guides...

Revision 1.10  2005/03/01 16:37:19  scmijg
Nearly finished Thin Client...

Revision 1.9  2005/03/01 15:43:50  scmijg
Slowly hacking user guide... near finished thin client init!

Revision 1.8  2005/03/01 15:07:59  scmijg
Slowly dumping out the user guide...

Revision 1.7  2005/03/01 12:53:59  scmijg
First screen shot added

Revision 1.6  2005/03/01 12:07:08  scmijg
More notes...

Revision 1.5  2005/02/28 22:40:47  scmijg
```

```
All version info files moved to java-test (so not carried around by release). Also means we can query
Interaction logging - now passes float, not int. Float represents time of initial request on client.

Revision 1.4  2005/02/28 16:29:42  scmijg
Getting there... :-)

Thin Client now launched!

Revision 1.3  2005/02/28 12:52:13  scmijg
Slowly working on docs...

Revision 1.2  2005/02/28 10:20:06  scmijg
Fixed the odd typo, LaTeX now builds again.

Revision 1.1  2005/02/28 10:01:33  scmijg
Renamed files to make life easier! Now all consistent...
Also added new "part" file for Architecture

Revision 1.1  2005/02/25 10:59:25  scmijg
Ok, now working with developer/install/user guides
```

References

[1] Apache. Apache Jakarta Project—Tomcat. http://jakarta.apache.org/tomcat/index.html.

[2] Apache. Apache Web Services Project—Axis. http://ws.apache.org/axis/index.html, 2004.